Domain Decomposition Methods for Nuclear Reactor Modelling with Diffusion Acceleration

Blake, Jack

Award date:
2016

Awarding institution:
University of Bath

Domain Decomposition Methods for Nuclear Reactor Modelling with Diffusion Acceleration

submitted by

Jack C. H. Blake

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

March, 2016

Summary

In this thesis we study methods for solving the *neutron transport equation* (or *linear Boltzmann equation*). This is an integro-differential equation that describes the behaviour of neutrons during a nuclear fission reaction. Applications of this equation include modelling behaviour within nuclear reactors and the design of shielding around x-ray facilities in hospitals. Improvements in existing modelling techniques are an important way to address environmental and safety concerns of nuclear reactors, and also the safety of people working with or near radiation.

The neutron transport equation typically has seven independent variables, however to facilitate rigorous mathematical analysis we consider the monoenergetic, steady-state equation without fission, and with isotropic interactions and isotropic source. Due to its high dimension, the equation is usually solved iteratively and we begin by considering a fundamental iterative method known as *source iteration*. We prove that the method converges assuming piecewise smooth material data, a result that is not present in the literature. We also improve upon known bounds on the rate of convergence assuming constant material data. We conclude by numerically verifying this new theory.

We move on to consider the use of a specific, well-known diffusion equation to approximate the solution to the neutron transport equation. We provide a thorough presentation of its derivation (along with suitable boundary conditions) using an asymptotic expansion and matching procedure, a method originally presented by Habetler and Matkowsky in 1975. Next we state the method of *diffusion synthetic acceleration* (DSA) for which the diffusion approximation is instrumental. From there we move on to explore a new method of seeing the link between the diffusion and transport equations through the use of a block operator argument.

Finally we consider domain decomposition algorithms for solving the neutron transport equation. Such methods have great potential for parallelisation and for the local application of different solution methods. A motivation for this work was to build an algorithm applying DSA only to regions of the domain where it is required. We give two very different domain decomposed source iteration algorithms, and we prove the convergence of both of these algorithms. This work provides a rigorous mathematical foundation for further development and exploration in this area. We conclude with numerical results to illustrate the new convergence theory, but also solve a physically-motivated problem using hybrid source iteration/ DSA algorithms and see significant reductions in the required computation time.

Acknowledgements

Over the course of writing this thesis I have had the pleasure of knowing and working with so many amazing and lovely people.

Firstly, my greatest thanks go to my two fantastic supervisors Ivan Graham and Alastair Spence. Your vast knowledge, dedication, guidance and patience led to the best PhD experience I could have hoped for. It continues to amaze me that no matter how many demands were on your time, there was always a meeting scheduled or time for a chat by the whiteboard. Despite the inevitable ups and downs, your commitment to me and to the project ensured that I (almost) always remained positive, and I will forever be deeply grateful for everything you have done.

A further huge thank you goes to my industrial supervisor Paul Smith from AMEC Foster Wheeler. Your expert knowledge and boundless enthusiasm were greatly motivational, and I feel lucky to have been able to work with you and everyone else in the fantastic ANSWERS team. In particular I would also like to thank Brendan and Chris, as well as fellow PhDers Mark and Dan, for their support and patient help. Huge thanks also go to Caroline Middlemas who made sure I was welcomed and involved at all times, and even drove me in to work on my first visit!

Importantly my thanks also go to The University of Bath, EPSRC and AMEC Foster Wheeler for generously funding my research.

I was lucky to have many many brilliant friends and colleagues who helped and supported me along the way, and I am thankful to them all. I will mention just a few here. First thank you to Doug and Elvijs, you were both hugely helpful and motivational in very different ways. A special thanks go to Steve, Katy, Amy and Jen who always made my day brighter and continue to do so whenever I see them. Also, thank you Marcus for your help at every hour of the day, and for joining Katy and me in "team coffee". Lastly to David: thank you for being an awesome friend and for every bit of support, both academically and beyond.

A huge thank you goes to my amazing family, and especially to my parents who have always believed in me however misplaced I often thought that belief was. I know this thesis would not have been possible without your unending love and support, and I do not say often enough how amazingly grateful I am to have you.

Lastly I thank my beautiful and perfect fiancée, Tara, for literally everything. Tara, you are my constant source of joy and happiness. You are supportive and loving, you make me laugh and you make me dream. I cannot imagine a more brilliant small. You are my life-person.

Contents

1 Introduction **4**

1.1 The Neutron Transport Equation in 3D 5

1.1.1 Simplifying Assumptions 8

1.1.2 Boundary Conditions 10

1.2 Solution Methods 11

1.2.1 Discrete Ordinates 12

1.2.2 Spherical Harmonics 12

1.2.3 Diffusion Approximation 12

1.2.4 Finite Element Methods 13

1.3 Literature Review 14

2 Source Iteration **19**

2.1 Introduction 20

2.2 Simplified Transport Equation 21

2.2.1 Transport Equation in 3D 21

2.2.2 Transport Equation in 2D 22

2.2.3 Transport Equation in 1D 23

2.2.4 Notation and Space 24

2.3 Solution to the Transport Equation in the Absence of Scattering 25

2.3.1 Formulae for Angular and Scalar Fluxes 25

2.3.2 Zero Boundary Conditions: the operator $\mathcal{K}_{\sigma_T}$ 31

2.3.3 Properties of the operator, $\mathcal{K}_{\sigma_T}$ 32

2.4 Source Iteration Algorithm 36

2.5 Convergence of Source Iteration 37

2.5.1 Preliminary Results 38

2.5.2 Convergence of Source Iteration 44

2.5.3 Improved Bounds in the case of Constant Cross Sections 46

2.6 Benefits and Limitations of Source Iteration 52

2.7 Numerical Results . . . 53
2.7.1 Verifying Theorem 2.23 . . . 54
2.7.2 Verifying Theorem 2.21 . . . 57

3 The Diffusion Approximation 60
3.1 Introduction . . . 60
3.2 Asymptotic Derivation of the Diffusion Equation and Appropriate Boundary Conditions in 1D . . . 62
3.2.1 Nondimensional Transport Equation . . . 63
3.2.2 Outer Expansion: expanding away from the boundaries . . . 65
3.2.3 Inner Expansion: expanding near to the boundaries . . . 70
3.2.4 Matching the Inner and Outer Expansions . . . 74
3.2.5 Boundary Conditions for the Diffusion Approximation . . . 75
3.2.6 Summary . . . 77
3.3 Diffusion Synthetic Acceleration . . . 79
3.3.1 Synthetic Acceleration . . . 79
3.3.2 DSA Algorithm . . . 83
3.3.3 DSA as a Preconditioner . . . 86
3.4 Block Operator Diffusion . . . 88
3.5 Numerical Results . . . 101
3.5.1 Comparing SI and DSA . . . 102
3.5.2 DSA in the Epsilon Limit . . . 106

4 Domain Decomposition 111
4.1 Background and Motivation . . . 111
4.2 Domain Decomposed Source Iteration (DDSI) . . . 113
4.2.1 Source Iteration Recap . . . 114
4.2.2 Jacobi DDSI . . . 115
4.2.3 Gauss-Seidel DDSI . . . 119
4.3 Convergence of Gauss-Seidel DDSI . . . 124
4.4 Convergence of Jacobi DDSI . . . 130
4.4.1 Bounding the Solution Operator Norm . . . 131
4.4.2 Convergence of Jacobi DDSI . . . 138
4.5 Numerical Tests . . . 145
4.5.1 Verifying Jacobi DDSI Convergence . . . 146
4.5.2 Verifying Gauss-Seidel DDSI Convergence . . . 148
4.5.3 Physically Motivated Example: Spent Fuel Pool . . . 153

5 Finite Element Method for the Steady Neutron Transport Equation 160
5.1 Introduction 160
5.2 Discontinuous Galerkin Finite Element Method 161
5.2.1 The Steady Neutron Transport Equation with Constant Angle . 161
5.2.2 The Steady Neutron Transport Equation 164
5.2.3 Error Calculation 167

6 Summary and Directions for Future Research 169

A Appendices 173
A.1 Solving a Homogeneous Integro-Differential Equation via Separation of Variables 173
A.2 Fourier Integral Theorem and other results 177

Chapter 1

Introduction

Contents

1.1 The Neutron Transport Equation in 3D 5
1.1.1 Simplifying Assumptions 8
1.1.2 Boundary Conditions 10
1.2 Solution Methods 11
1.2.1 Discrete Ordinates 12
1.2.2 Spherical Harmonics 12
1.2.3 Diffusion Approximation 12
1.2.4 Finite Element Methods 13
1.3 Literature Review 14

A central problem in applied nuclear physics is that of accurately and efficiently modelling the distribution of neutrons within a nuclear reactor. This distribution is very closely modelled by the *neutron transport equation* (also called the *linear Boltzmann equation*), which is an integro-differential equation with typically seven independent variables. It is only after many simplifying assumptions are made that this can be solved exactly. Instead for almost all useful applications this equation is solved numerically, and there are a variety of different ways this can be done, as we will discuss in Section 1.2.

As well as in nuclear reactor modelling, the neutron transport equation is important in so-called *shielding* calculations. These are used whenever radiation from a radioactive source needs to be prevented from leaking into the surrounding area. Applications include the design of shielding around x-ray facilities in hospitals as well as for dry and wet storage of spent nuclear fuel.

The long-term use of nuclear power as a source of clean and reliable energy is sometimes uncertain. Nonetheless, working to improve the accuracy, efficiency and

versatility of our existing modelling techniques is an important way to address environmental and safety concerns. This includes not only the safety and efficiency of new and existing nuclear reactors, but also the safety of people working with or near radiation in any significant form.

The work in this thesis was conducted during a PhD project at the Universiy of Bath in liason with the ANSWERS® Software Service, AMEC Foster Wheeler.

We start this introduction in Section 1.1 by introducing the neutron transport equation along with the most commonly used boundary conditions. This equation will be our focus for the majority of this thesis, however first we will make several simplifying assumptions in order to facilitate the later mathematical analysis. These assumptions are specified in Section 1.1.1, and will be used to give simplified versions of the neutron transport equation in 3D, 2D and 1D at the start of Chapter 2. In Section 1.2 we will give a quick overview of some of the main solution methods from the literature. Lastly, in Section 1.3 we will broadly outline the work carried out in this thesis and review the related state-of-play in existing literature. We will then specify the contributions that will be made by this thesis.

1.1 The Neutron Transport Equation in 3D

As we mentioned above, the neutron transport equation models the behaviour of neutrons within a nuclear fission reaction. It is derived by carefully considering how a quantity called the *angular neutron density*, often denoted $N(\mathbf{r}, \Omega, E, t)$, changes in time. This quantity represents the number of neutrons in a unit volume at position $\mathbf{r} \in V \subset \mathbb{R}^3$ travelling in direction $\Omega \in \mathbb{S}^2$ with kinetic energy $E \in \mathbb{R}^+$ at time $t \in \mathbb{R}^+$. Detailed descriptions of this derivation can be found in [57, Chapter 1], [20], [69], [24], [12, Section 1] and [59, Section 2]. The main quantity of interest in the neutron transport equation is called the *(angular) neutron flux* (or *fluence rate*) and is related to the angular neutron density via

$$\psi(\mathbf{r}, \Omega, E, t) = v(E)N(\mathbf{r}, \Omega, E, t), \tag{1.1}$$

where $v(E) = \sqrt{2E/m}$ is the neutron speed (m being mass). We can obtain a physical interpretation for the angular neutron flux by considering a small, dimensionless subset of the domain, $\mathrm{d}V\mathrm{d}\Omega\mathrm{d}E$, about $(\mathbf{r}, \Omega, E)$ at time t. With this we can say that in an increment $\mathrm{d}t$ at time t, $\psi\mathrm{d}V\mathrm{d}\Omega\mathrm{d}E\mathrm{d}t$ is the total distance (or *path length*) travelled by neutrons in $\mathrm{d}V\mathrm{d}\Omega\mathrm{d}E$ about $(\mathbf{r}, \Omega, E)$ during the time interval $\mathrm{d}t$ at time t.

To derive the neutron transport equation, it is assumed that neutrons are point particles and consequently neutron-neutron interactions are neglected (see [57, Section

1-2]). Full knowledge of the material properties of the domain is also assumed. Under these assumptions, the neutron transport equation is derived by carefully considering the rate of change of the neutron flux in time, $\psi(\mathbf{r}, \Omega, E, t)$, determined by the difference between its rate of gain and rate of loss (see e.g. [59, Section 2.5]). Before stating the equation, we will talk through the various events that can lead to these losses and gains.

As mentioned, the neutron transport equation describes the behaviour of the neutron flux, $\psi(\mathbf{r}, \Omega, E, t)$. It bases this description upon the likelihood of various neutron interactions (or *collisions*) occurring, and based on the characteristics of a *neutron source*. When modelling a nuclear reactor it is generally specified that neutrons can undergo three types of interaction: they can cause *fission*, can be *scattered* or they can be *captured*. We consider the three interactions in order.

First of all, a neutron could collide with some *fissile material* and initiate a *fission event*. Fissile material is material which, upon collision with a low-energy (*slow* or *thermal*) neutron, can capture it and then undergo a fission event [69]. This releases a number of new neutrons, specified by $\nu(E) \in \mathbb{R}^+$ for a collision caused by a neutron with energy E. This is a *fission collision* and the likelihood of such a collision occurring is denoted by the variable $\sigma_F(\mathbf{r}, E) \in \mathbb{R}^+$, known as the *fission cross section*. The neutrons produced by the fission may be travelling in any direction with no bias (i.e. they are *isotropic* in angle) regardless of the direction of travel of the colliding neutron, and so angle of travel is not a consideration for this type of collision. However, the neutrons are released over a spectrum of different energies specified by $\chi(E) \in \mathbb{R}^+$, where $\chi(E)\mathrm{d}E$ is the probability that a neutron produced during fission will have an energy within $\mathrm{d}E$ of E [57]. This leads to the energy dependence of the fission cross section.

Next, upon collision with a nucleus, a neutron could be deflected and so end up travelling in a different direction with different energy. In this case the neutron is said to have been *scattered* and the likelihood of such an event occurring is denoted by the variable $\sigma_S(\mathbf{r}, \Omega' \cdot \Omega, E', E) \in \mathbb{R}^+$, known as the *scatter cross section*. Here the neutron is scattered from travelling with energy E' in direction Ω' to travelling with energy E in direction Ω. This interaction is rotationally invariant, and as a result the cross section depends only on the cosine, $\Omega' \cdot \Omega$.

Lastly, upon collision with a nucleus, a neutron could be *captured* and so no longer be considered within the ongoing reaction. The likelihood of such an event occurring is denoted by the variable $\sigma_C(\mathbf{r}, E) \in \mathbb{R}^+$ and is known as the *capture cross section*. If we denote by $\sigma_T(\mathbf{r}, E) \in \mathbb{R}^+$ a quantity known as the *total cross section*, defined to be the likelihood of any collision occurring to neutrons at position $\mathbf{r}$ with energy E, then

the following relation holds

$$\sigma_T(\mathbf{r}, E) = \sigma_F(\mathbf{r}, E) + \frac{1}{4\pi}\int_{\mathbb{R}^+}\int_{\mathbb{S}^2} \sigma_S(\mathbf{r}, \Omega \cdot \Omega', E, E')\, \mathrm{d}\Omega'\, \mathrm{d}E' + \sigma_C(\mathbf{r}, E). \tag{1.2}$$

Here we want to include neutrons that are scattered into all other angles and energies, and so the scattering cross section is integrated over all possible outgoing states. We will also define for convenience $\sigma_A(\mathbf{r}, E) \equiv \sigma_C(\mathbf{r}, E) + \sigma_F(\mathbf{r}, E)$. This is called the *absorption* cross section, and represents all collisions which result in the neutron being absorbed.

It is important to note that since $\psi(\mathbf{r}, \Omega, E, t)$ only considers specific angles and energies, after each of the three collision types the neutron is no longer travelling in the same direction with the the same energy, and so is no longer a part of that specific neutron flux.

We will also include a *neutron source* term, which will be denoted by $Q(\mathbf{r}, \Omega, E, t)$ and is a non-fission source term of neutrons from position $\mathbf{r}$ with energy E in direction Ω at time t. This represents an emission of neutrons by some source, such as radioactive material, without the need for a fission event.

Using this notation we can now state the time-dependent 3D neutron transport equation as follows

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial}{\partial t}\psi(\mathbf{r}, \Omega, E, t) \;+\; \Omega \cdot \nabla\psi(\mathbf{r}, \Omega, E, t) + \sigma_T(\mathbf{r}, E)\psi(\mathbf{r}, \Omega, E, t) \;=\; & \\ \int_{\mathbb{R}^+}\int_{\mathbb{S}^2} \sigma_S(\mathbf{r}, \Omega' \cdot \Omega, E', E)\psi(\mathbf{r}, \Omega', E', t)\, \mathrm{d}\Omega'\, \mathrm{d}E' & \\ +\; \frac{\chi(E)}{4\pi}\int_{\mathbb{R}^+} \nu(\mathbf{r}, E')\sigma_F(\mathbf{r}, E')\int_{\mathbb{S}^2} \psi(\mathbf{r}, \Omega', E', t)\, \mathrm{d}\Omega'\, \mathrm{d}E' & \\ +\; Q(\mathbf{r}, \Omega, E, t), & \end{aligned} \tag{1.3}$$

for $(\mathbf{r}, \Omega, E, t) \in V \times \mathbb{S}^2 \times \mathbb{R}^+ \times \mathbb{R}^+$. This equation describes how the angular neutron flux, ψ, varies in time. The second and third terms on the left of (1.3) represent neutron loss from the system, while the terms on the right represent neutron gain.

We will now briefly talk through the physical meanings of each term, starting with the terms on the left. Firstly, the second term on the left is a *convection* term and represents the rate of change of neutrons in the considered space and energy due to *streaming*. In this context streaming refers to the motion of neutrons in a straight line without any collision occurring. The third term is a *sink* term, representing the loss of

neutrons that undergo any type of collision since they will now be travelling in a new direction with new energy.

The terms on the right of (1.3) are all *source* terms, representing the gain of neutrons through different physical mechanisms. The first term on the right adds in those neutrons which have been scattered from other energies and directions (E' and Ω') into the considered energy and direction (E and Ω) at time t, and so are now to be considered a part of the flux. To consider the in-scatter of neutrons from all other energies and directions, this term is integrated over the appropriate domains. The second term adds neutrons that have been produced by nuclear fission, travelling in the correct direction with the correct energy at time t. Finally $Q(\mathbf{r}, \Omega, E, t)$ adds in neutrons produced by the non-fission source.

This equation plays an important role in many different applications of nuclear reactors across an array of disciplines. These range from medical applications (such as the production of radio-isotopes and radiation therapy), through propulsion methods for ships and also its most well known application in nuclear power stations producing electricity [69]. Outside of nuclear reactors it is also solved in shielding calculations (see [59, Section 2.11.4] and elsewhere in [19]), such as are used for ensuring adequate safety measures around x-ray machines in hospitals. To fulfil these needs, a wide array of industrial modelling codes rely upon efficient and accurate solutions of the neutron transport equation. The transport equation, (1.3), is seven-dimensional, which necessitates the use of iterative methods to achieve accurate solutions within a reasonable amount of time. As a result much interest and ongoing research is focussed around improving the efficiency of these iterative methods, and it is on this topic that our work is based.

1.1.1 Simplifying Assumptions

To facilitate rigorous mathematical analysis, the Boltzmann transport equation is often simplified by making one or more assumptions. Indeed in this thesis we will only be considering a heavily simplified form of the transport equation, (1.3), given above. In this section we will carefully describe each of the simplifying assumptions that we will make and try to give an explanation of the physical impacts of each.

One of the most straightforward simplifications we make will be to consider the steady-state neutron transport equation. This allows us to remove the time dependence from all variables and to drop the partial time derivative on the left hand side. By doing this we will be considering the system at equilibrium, with an enforced balance between the loss and gain of neutrons. The time-independent form of the transport equation is often formed as an eigenvalue problem, in which the smallest real eigenvalue determines

the *criticality* of the system (see [59, Section 2.7], [67, Section 1.5], [69, Section 2.2]).

We will ignore energy dependence entirely, and instead focus on the so-called *one-speed* or *monoenergetic* version of the transport equation. On the surface this seems to be a drastic assumption to make since, in reality, the energy a particle has dramatically affects the cross sections. In fact the variation of the cross sections with respect to a neutron's energy is so complex that it cannot be calculated or accurately modelled at each point. Instead a range of energy intervals (or *groups*) are considered, leading to the so-called *multi-group equations* (see Lewis and Miller [57, Chapter 2], Prinja and Larsen [59, Chapters 2 and 6]). In practice within the multi-group treatment, solutions to monoenergetic equations are required and neutron transfers into one energy group from all others appears as a source term. Consequently work on the monoenergetic form of the neutron transport equation is still very relevant, and it continues to be the subject of research and analysis. Another benefit of the monoenergetic form is that it admits deeper mathematical analysis and, in certain geometries, allows analytic solutions to be found [59, Section 2.9].

We will assume that scattering interactions are isotropic in angle, i.e. the scattering cross section is independent of angle. To establish this independence we recall that the scattering interaction is rotationally invariant, and so the cross section can be written as an expansion in terms of Legendre polynomials. Defining these polynomials via Rodrigues' formula as

$$P_n(\mu) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}\mu^n} \left(\mu^2 - 1\right)^n, \quad n \in \mathbb{N}_0, \tag{1.4}$$

we can thus write the monoenergetic scattering cross section as

$$\sigma_S(\mathbf{r}, \Omega' \cdot \Omega) = \sum_{n=0}^{N} \frac{2n+1}{4\pi} \sigma_{S,n}(\mathbf{r}) P_n(\Omega' \cdot \Omega), \tag{1.5}$$

(see [59, Section 3.1]). In most applications this expansion is truncated, and if $N = 0$ is chosen then the scattering is said to be *isotropic*. This leaves

$$\sigma_S(\mathbf{r}) = \frac{1}{4\pi} \sigma_{S,0}(\mathbf{r}), \tag{1.6}$$

however since this thesis will be solely focussed on the case of isotropic scattering, we will simply use the notation σ_S and gain the $1/4\pi$ scale.

A seemingly large assumption that we will make will be to assume there are no fission interactions. By doing this we totally remove the fission term from the right hand side of (1.3), resulting in a much simpler equation of interest. While this does appear to be a big assumption, the fission term can be thought to have been included

implicitly in the scatter term with exactly one neutron being produced per collision. Because of this our later analysis can be extended to apply to both the with fission and without fission cases under certain assumptions, and indeed we will see a basic example of this in the later numerical results sections.

Lastly, while the source can include angular dependency, most natural source materials are isotropic in angle. Consequently we are physically justified in using a non-fission source term with no angular dependence.

Under these assumptions the transport equation (1.3) becomes

$$\Omega \cdot \nabla\psi(\mathbf{r},\Omega) + \sigma_T(\mathbf{r})\psi(\mathbf{r},\Omega) = \frac{\sigma_S(\mathbf{r})}{4\pi}\int_{\mathbb{S}^2} \psi(\mathbf{r},\Omega')\,\mathrm{d}\Omega' + Q(\mathbf{r}). \tag{1.7}$$

for $(\mathbf{r},\Omega) \in V \times \mathbb{S}^2$. This is the form of the linear Boltzmann transport equation that we will consider for the majority of this thesis. Occasionally we will need to make further assumptions, and they will be stated and explained when relevant. Whilst we will aim to work with the 3D form as much as possible, treatment of lower dimensional versions will be neccesary. These versions will be thoroughly defined in Section 2.2. We finish this set of assumptions by mentioning that for much of this thesis we will assume that the cross sections are piecewise smooth. This assumption means our convergence work in Chapter 2 allows for more general material properties that has been possible before. We will clarify this assumption when appropriate in Chapter 2.

1.1.2 Boundary Conditions

As with any differential equation, (1.7) can only be solved when combined with some relevant boundary conditions. These can take different forms and here we will consider two types: explicit boundary conditions through the imposition of a *boundary source* term, and implicit boundary conditions through a *reflecting* boundary requirement (see [59, Section 2.4], [57, Section 1-3] among others).

First of all we can explicitly specify an incoming neutron flux on the outer boundary of the spatial domain, ∂V. This boundary flux, say $f(\mathbf{r},\Omega)$, must be specified for all $\mathbf{r} \in \partial V$ and for all angles pointing in to the domain. These angles are found by requiring $\Omega \cdot n(\mathbf{r}) < 0$, where $n(\mathbf{r})$ denotes the outward unit vector, normal to the surface ∂V at $\mathbf{r}$. Therefore, to enforce this boundary condition we require that

$$\psi(\mathbf{r},\Omega) = f(\mathbf{r},\Omega) \quad \text{when } n(\mathbf{r})\cdot\Omega < 0, \;\; \forall \mathbf{r} \in \partial V. \tag{1.8}$$

A special case of these are *vacuum* (or *zero*) boundary conditions with $f = 0$ to impose a requirement for zero incoming neutron flux. This means that the only source of neutrons under consideration is from within the reactor itself, and we impose

$$\psi(\mathbf{r}, \Omega) = 0 \quad \text{when } n(\mathbf{r}) \cdot \Omega < 0, \ \ \forall \mathbf{r} \in \partial V. \tag{1.9}$$

Next, reflecting boundary conditions specify that the incoming flux and outgoing flux at the boundary of the spatial domain are equal, and are imposed as follows. For all points $(\mathbf{r}, \Omega_1) \in \partial V \times \mathbb{S}^2$, define the *local reflection* to be the reflection of Ω_1 in the tangent plane of ∂V at $\mathbf{r}$. If Ω_2 is the local reflection of Ω_1, then

$$\psi(\mathbf{r}, \Omega_1) = \psi(\mathbf{r}, \Omega_2). \tag{1.10}$$

This ensures that the boundary flux in any outgoing angle equals that of the reflected incoming angle at each point on the boundary of the domain. This prevents neutrons from escaping the system, instead *reflecting* them back inwards. In more complicated geometries these can be used to model infinite arrays of a certain region. This is done by defining one copy of the region and then applying reflecting conditions around the boundary. This implicitly assumes an average flux of zero over boundaries between the regions, however it can still provide a useful representation of the set up.

1.2 Solution Methods

In this section we will talk about some of the different types of methods that are used to solve the transport equation in modelling nuclear fission reactors. Broadly these methods can be broken down into two genres: *deterministic* methods and *Monte Carlo* (stochastic) methods. Monte Carlo methods are discussed widely in the literature (see [57, Chapter 7], [24, Chapter 9]) and are currently used to model reactor criticality. These methods have an advantage in that they do not depend upon meshing the domain, and so the complexity of the domain does not dramatically affect the solve time. For this reason they are often preferred for modelling complex geometrical set-ups. The majority of our analysis will be focussed around deterministic methods, and in this section we will give a short overview of some of the main deterministic methods that are used in the literature.

Deterministic methods are those that will always produce the same output for a given input. These methods discretise the transport equation and form a system of coupled algebraic equations which can then be solved. This can involve using iterative methods (provided they don't contain calls to random variables), such as *Krylov* methods (see [39], [65], [66]). Deterministic approaches include methods like *discrete ordinates* (the S_N-method), *spherical harmonics* (the P_N-method) and *diffusion approximation*.

1.2.1 Discrete Ordinates

The *discrete ordinates* method works by sampling the angular variable at a number of discrete points, and then replacing integrals over angle by weighted quadrature summations. The quadrature points and weights should be chosen so that all directions of the neutron flux are given equal importance, and further detail and an example is given in Chapter 5. This yields a semidiscrete system of equations, which can then be discretised in space via some finite difference or finite element method, allowing a numerical solution method to be applied. This method is often called the S_N-method, and further information on it can be found in [70, Chapter 4], [57, Chapters 3 and 4], [69, Chapter 9], and [32, Chapter 9] among others.

1.2.2 Spherical Harmonics

The *spherical harmonics* method for solving the neutron transport equation works by expanding the angular component of the neutron flux in terms of spherical harmonics. By truncating this expansion a finite system of semidiscrete equations is obtained, which can be further simplified using orthogonality. Discretising the spatial variable using finite difference methods or finite element methods then allows for the application of a suitable solver. This method is often called the P_N-method, and further information can be found in [12, Chapter 3] or [57, Chapter 3].

1.2.3 Diffusion Approximation

The fundamental idea behind this solution method is that, under certain conditions, the quantity

$$\phi(\mathbf{r}) \equiv \frac{1}{4\pi}\int_{\mathbb{S}^2} \psi(\mathbf{r}, \Omega)\, \mathrm{d}\Omega,$$

known as the *scalar flux* can be approximated well by the solution to a specific diffusion equation. This diffusion equation is given by

$$-\frac{1}{3\sigma_T}\nabla \cdot \left(\frac{1}{3\sigma_T(\mathbf{r})}\nabla\Theta(\mathbf{r})\right) + \sigma_A(\mathbf{r})\Theta(\mathbf{r}) = Q(\mathbf{r}), \tag{1.11}$$

along with appropriate boundary conditions. This equation and its boundary conditions will be rigorously derived (in 1D) in Chapter 3 by following the work in [34]. More information on diffusion theory in the context of neutron transport can be found in Prinja and Larsen [59, Chapter 8], Stacey [69, Chapter 3], Duderstadt and Martin [24, Section 4.2], Bell and Glasstone [12, Secion 3.1] or Tait [70, Chapter 5].

To make this approximation several assumptions are made (see for example [54], [69, Chapter 3]), which will be given in Chapter 3. Unfortunately these assumptions do not hold near material interfaces and domain boundaries, but the approximation can still give accurate predictions by working on a homogenised domain where the cross sections are averaged spatially (see [69, Chapter 3, p.47], [59, Section 8.4]) and by working with the more accurate transport theory to ensure accuracy.

Extension of this theory into an iterative acceleration scheme, known as *diffusion synthetic acceleration*, will also be addressed in Chapter 3. Good information on this can be found in Adams and Larsen [2, Chapters 1 and 2], and Lewis and Miller [57, Section 2-4, p.97], among others.

1.2.4 Finite Element Methods

Finite element methods are a class of discretization method used to approximate the solution to partial differential equations subject to boundary constraints. Generally speaking, they work by subdividing the domain into a set of small pieces (or *elements*) and then solving a simpler local problem on each element. These local solutions can then be combined to obtain a finite dimensional approximation to the true solution over the whole domain. The local problems are found by taking the weak formulation of the PDE on each element and choosing test functions from a set of basis functions (often piecewise-polynomial 'hat' or 'tent' functions are used). This process eliminates the spatial derivatives and for steady-state problems results in a set of algebraic equations.

Finite element methods are widely used within the nuclear industry, and are a dominant method in industry in general. Their application to neutron transport problems began in the early 1970s, however it was only after available computer memory grew rapidly that they began to be used within industrial codes [11, Chapter 2]. Large amounts of industrial and academic research is devoted to their development. Though the basic method is fairly straightforward there are many different choices to make, including the choice of elements and basis functions. These choices are often problem-dependent and have a big effect on the accuracy of the method. For detailed information on the theory and application of finite element methods see Brenner and Scott [16] or Grossmann et. al [33, Chapter 4].

To conduct numerical experiments throughout this thesis we used discontinuous Galerkin finite elements to discretise the neutron transport equation. This process is detailed in Chapter 5, where references specific to neutron transport are given. We also used continuous finite elements to discretise the diffusion equation (see Chapter 3) however we do not explain this in detail and instead refer the reader to Brenner and Scott [16], for example.

1.3 Literature Review

In this section we will discuss and review existing literature that is relevant to the work that has been completed in this thesis. We will roughly follow the order in which work will appear over the coming chapters. To conclude the section we will outline the specific contributions that this thesis makes to the literature.

One of the most basic iterative methods used to solve the neutron transport equation is known as *source iteration*, though it is equivalent to Richardson iteration, a method proposed by L. Richardson in 1910 [62]. Iterative solvers are used in current industrial deterministic software for solving criticality problems including the neutron transport problem. One example is in the WIMS code (part of the ANSWERS® Software Service provided by AMEC Foster Wheeler) where two of the deterministic methods mentioned in Section 1.2 can be used (S_N and diffusion theory) as well as the method of characteristics and the method of *collision probabilities* (more information on both of these methods within a nuclear physics setting can be found in [19] and [69]), alongside iterative solvers. They are still a very relevant part of the industry and as such any improvements in efficiency (and so accuracy) of solves can only be beneficial for both energy and environmental concerns alike: firstly for reducing associated risks, and secondly for continuing to satisfy growing energy demands [69, Preface]. Currently most industrial level nuclear modelling software packages use source iteration to some extent, which we will introduce in Chapter 2. Information on source iteration can be found in [2], [32], [67], and many others. Various methods of acceleration, including Krylov methods, are utilised to speed up the convergence of this method and so work in this area is still very relevant.

In [26], T. M. Evans et. al. present a new method called Monte Carlo synthetic acceleration (MCSA) and use it to solve a radiation diffusion equation for the scalar flux. This method builds upon source iteration, and they demonstrate that for certain model problems MCSA can outperform standard solution techniques such as CG and GMRES. Furthermore an important issue in modern computing where massively parallel applications are becoming increasingly common robust methods require resiliency to hardware failures that occur during a solve. T. M. Evans et. al. argue that MCSA can provide a good basis for such resiliency, lending support the relevance of source iteration as part of modern iterative methods.

In Chapter 2 we derive new convergence theory for source iteration. We will present a new tighter bound on the convergence of source iteration for constant cross sections, as well as presenting a new convergence result that allows for piecewise smooth cross sections. Later in Section 2.7 we will carry out numerical tests to support our theory.

We will see that source iteration performs well when applied to systems in which particles typically undergo only a small number of interactions before being captured. On the other hand we will observe that in systems where particles undergo a large number of interactions before being captured, source iteration converges slowly. As explained in [2, Section I.B], such systems are characterised by scattering dominated interactions within *optically thick* domains (where the average distance between successive collisions is small compared to the domain width). Under these conditions a domain is said to be *diffusive*, and we are motivated to seek some way of improving the performance of source iteration. One way of doing this is examined in Chapter 3.

In the neutron transport literature it is well known that a certain diffusion equation can provide a good approximation to the scalar flux within a diffusive domain (in particular, see [59, Chapter 8], [69, Chapter 3], [2, Chapters 1 and 2], [70, Chapter 5], [54] and good boundary condition discussion can be found in [34], with the discrete case looked at in [40], [41] and [29]). In the literature, several different methods are used to derive this equation. One method works by writing the neutron flux as a Legendre polynomial expansion and then truncating after the first two terms. The resulting approximate diffusion equation is commonly referred to as the P_1-approximation, with a more accurate P_N-approximation following by truncating the expansion later.

In Chapter 3 we derive this diffusion equation following an asymptotic expansion method used by Habetler and Matkowsky [34], though our work differs slightly as we treat the non-fission source term explicitly. One advantage of this method is that suitable boundary conditions can be found through a subsequent boundary layer analysis. Other methods of obtaining suitable boundary conditions, such as requiring the diffusion equation to be valid uniformly up to the boundary, do not obtain the correct conditions (see [40] and [29]).

This diffusion approximation can be used to construct an acceleration scheme known as *diffusion synthetic acceleration* (or DSA). This works in conjunction with another iterative scheme (such as source iteration) and uses diffusion solves to update the approximate solution at each iteration. See [2, Chapters 2 and 3] and [7] for analysis; see [5] and [51] for application. A lot of effort has also been put into overcoming numerical instability issues caused by certain discretisations of the transport and diffusion equations (see [5] for the recognised solution, [51] for some follow up work), and this has led to a good understanding of how to guarantee stability in diffusion synthetic acceleration schemes.

Acceleration schemes can be shown to be equivalent to *preconditioning* methods ([32] and [2, Chapter 1] for discussion). Preconditioning a system involves transforming it (via some process or operator known as a *preconditioner*) into a form more easily

solved using iterative methods. The key to this is that the cost of preconditioning should not outweigh the benefits. Preconditioning is a frequently used process, backed up with a wealth of mathematical knowledge and understanding (see [32], [66] or [14]). Preconditioning in a transport-specific setting is considered in Adams and Larsen, [2]. DSA can be formulated as a preconditioner, and this is done in both Faber and Manteuffel, [27], and Ashby et. al., [6]. Ashby et. al. then apply DSA as a preconditioner to source iteration as well as to the Krylov-subspace method GMRES (though Brown [17] gives a more thorough account of the required discretisation). Later Warsa et. al. ([74], [75]) show that in the presence of material discontinuities, multidimensional DSA suffers a degradation in effectiveness. They demonstrate that by applying DSA as a preconditioner to a Krylov method this degradation can be avoided.

DSA converges more quickly than source iteration, particularly in diffusive domains, however it is also computationally more expensive (per iteration) to implement. As a result for some problems, though DSA may need fewer iterations to converge, in reality source iteration might still be the faster method (see Adams and Larsen, [2, Section II.B], for a discussion of this cost balance). Alternatively source iteration might converge slowly due to only a small part of a domain exhibiting diffusive behaviour. In this situation it would be beneficial to apply DSA only in that small part, and to apply source iteration over the rest of the domain. This provides the motivation behind our work in Chapter 4, in which we develop two domain decomposition source iteration algorithms (Algorithms 6 and 8). These algorithms can be applied to 3D spatial domains decomposed into an arbitrary number of subdomains. We go on to prove that both algorithms converge (Theorem 4.8 and Theorem 4.4) and in particular that Algorithm 8 is equivalent to the full source iteration algorithm.

Domain decomposition methods have been applied to solving the neutron transport equation before, both in space (see Yavuz and Larsen [76], [77]) and in angle (see Y. Azmy [8]). In particular, Yavuz and Larsen developed and implemented a domain decomposition source iteration algorithm focussed towards parallelisation. They demonstrated the speed-up gained by applying it over varying numbers of processors, with the number of subdomains always equalling the number of processors. The method they developed is similar to the *Jacobi* domain decomposed source iteration algorithm we present in Section 4.2.2, though they differ in when the subdomains communicate. They also state (though do not implement) a semi-discrete domain decomposed source iteration algorithm for rectangular grids of subdomains. This is a form of the *Gauss-Seidel* domain decomposed source iteration algorithm that we present in Section 4.2.3, however we work in a continuous setting and allow for decomposition into any number of convex subdomains. More recently Gonçalves and Coelho [30] considered parallel

algorithms for the discrete ordinates method. They implemented the parallel-focussed method of Yavuz and Larsen, however they did not attempt to analyse the method theoretically. We do not know of any other work developing the algorithms of Yavuz and Larsen, or in particular of any work proving convergence of the two different algorithms in a fully continuous setting as we do in Sections 4.3 and 4.4.

To carry out numerical tests at the end of each chapter, we discretised the neutron transport equation using discontinuous Galerkin (DG) finite elements in space and discrete ordinates in angle. This process is detailed in Chapter 5 and is intended to be used for reference to understand more fully the numerical testing that was carried out. DG methods were originally proposed in the early 1970s for solving partial differential equations. In 1973, Reed and Hill [61] published a DG method for solving the neutron transport equation. Following this, LaSaint and Raviart [56] covered the spatial discretisation in detail, and established uniqueness and existence of the resulting approximate solution. Later Johnson and Pitkäranta [43] derived a sharp bound on the error of DG finite element methods using nth degree polynomial basis functions. At the time it was not know that their bound was sharp, however this was demonstrated in 1991 by Peterson [58] using a counter example.

We conclude this introductory chapter by outlining the contributions that will be made by this thesis to the literature.

- We will provide new convergence theory for source iteration which proves convergence assuming piecewise smooth cross sections (Section 2.5.2).
- We will improve upon the existing convergence theory for source iteration assuming constant cross sections (Section 2.5.3).
- We will thoroughly present the asymptotic expansion derivation of the diffusion equation, and the boundary layer analysis required to obtain suitable boundary conditions, originally presented by Habetler and Matkowsky [34]. This is not original work, but is carefully and thoroughly presented to provide a useful addition to the existing literature (Section 3.2).
- We explore a new block operator approach to seeing the link between the diffusion approximation and the transport equation (Section 3.4).
- We present two different domain decomposed source iteration (DDSI) algorithms which provide various advantages (as well as disadvantages) to solving the neutron transport equation (Section 4.2).

- We prove convergence of both of these DDSI algorithms, and so provide a rigorous mathematical foundation for further development and exploration in this area (Sections 4.3 and 4.4). These are new results.

Wherever possible we provide numerical results to illustrate and support our theory, as well as to highlight any limitations. These results are provided in the last section of each chapter, and the finite element discretisation required to carry out the experiments is explained in Chapter 5.

Chapter 2

Source Iteration

Contents

2.1 Introduction 20

2.2 Simplified Transport Equation 21

2.2.1 Transport Equation in 3D 21

2.2.2 Transport Equation in 2D 22

2.2.3 Transport Equation in 1D 23

2.2.4 Notation and Space 24

2.3 Solution to the Transport Equation in the Absence of Scattering 25

2.3.1 Formulae for Angular and Scalar Fluxes 25

2.3.2 Zero Boundary Conditions: the operator $\mathcal{K}_{\sigma_T}$ 31

2.3.3 Properties of the operator, $\mathcal{K}_{\sigma_T}$ 32

2.4 Source Iteration Algorithm 36

2.5 Convergence of Source Iteration 37

2.5.1 Preliminary Results 38

2.5.2 Convergence of Source Iteration 44

2.5.3 Improved Bounds in the case of Constant Cross Sections . . . 46

2.6 Benefits and Limitations of Source Iteration 52

2.7 Numerical Results 53

2.7.1 Verifying Theorem 2.23 54

2.7.2 Verifying Theorem 2.21 57

2.1 Introduction

In this chapter we concern ourselves with understanding the most basic iterative method for solving the neutron transport equation, known as *source iteration*. We will prove a new result regarding the convergence of source iteration under more general assumptions than has been achieved before in a continuous setting. More precisely, our result will apply when the cross sections are piecewise smooth. This situation is very relevant physically since cross sections govern how neutrons interact with the material they are travelling through. Most physical systems that are modelled contain more than one material and typically the boundary between these materials is distinct and thus causes a 'jump' in the cross section in space. Therefore results concerning piecewise smooth cross sections are more relevant than their constant counterparts.

A similar result with spatially dependent cross sections is proved by Ashby et. al [7, Section 4] in a special discrete case. This motivated us to consider a proof for the non-discretised problem, which allows the result to be used as a starting point for understanding the convergence in any discretisation. Consequently our work is a useful addition to the literature in understanding source iteration.

The method of source iteration itself is still a very relevant method in industry today. Recently it has been used by Evans et. al [26] as part of a Monte Carlo synthetic acceleration method. This method is aimed at being a *robust* iterative solver which can cope with the occurrence of hardware failures during massively parallel iterations.

We begin in Section 2.2 by defining the transport equation in 3, 2 and 1 spatial dimensions under the simplifying assumptions that will be required for our convergence result. Next in Section 2.3 we derive the solution to the transport equation without scattering and with homogeneous boundary conditions in each dimension with piecewise smooth total cross section. With these solutions we then define an associated solution operator, denoted $\mathcal{K}_{\sigma_T}$, under the further assumption of zero boundary conditions in each dimension. This operator has been considered before (see [45], [67, Chapter 2], [24, Chapter 2]) under the more restrictive assumption of constant cross sections. We will prove several useful properties of this operator, and in particular will show that it is *positive-definite* (see Lemma 2.13). In Section 2.4 we will introduce the source iteration algorithm, which will then be the subject of our convergence analysis in Section 2.5. The main convergence result will require preliminary work that will be covered in Section 2.5.1, and using that work our main convergence result, Theorem 2.21, will follow in Section 2.5.2. Next in Section 2.5.3 we will consider the case of constant cross sections, and will prove new tighter bounds on the rate of convergence for source iteration in 3D, 2D and 1D. Lastly we will consider the implications of our theory, and in Section 2.7

will give some numerical results that highlight the benefits and limitations of source iteration.

2.2 Simplified Transport Equation

In this section we will specify the transport equation in 1, 2 and 3 spatial dimensions in the forms that will be used throughout this chapter. We will use the simplifying assumptions explained in Section 1.1.1, which were the following. Firstly we consider only one energy group and so energy does not feature in our version of the transport equation. We also consider the equation at steady-state and so there is no time dependency. Lastly, we assume that the cross sections are isotropic (i.e. independent of angle) and piecewise smooth in space. As mentioned in the introduction, this last assumption means our convergence result applies to more physically relevant cross sections than has been possible before. We start by working in three spatial dimensions.

2.2.1 Transport Equation in 3D

We will perform much of the work in this chapter in three dimensions and will only work otherwise when the extension to lower dimensions is not straightforward. Recall from Chapter 1, in 3D the transport equation governs the behaviour of the *neutron flux*, denoted $\psi(\mathbf{r}, \Omega)$. Here $\mathbf{r} \in V$ is a spatial variable, where V is some bounded domain in $\mathbb{R}^3$, and $\Omega \in \mathbb{S}^2$ is an angular variable. Under the assumptions restated above, the monoenergetic, steady-state neutron transport equation with isotropic interactions and an isotropic neutron source is defined as

$$\Omega \cdot \nabla\psi(\mathbf{r}, \Omega) + \sigma_T(\mathbf{r})\psi(\mathbf{r}, \Omega) = \frac{\sigma_S(\mathbf{r})}{4\pi} \int_{\mathbb{S}^2} \psi(\mathbf{r}, \Omega)\, \mathrm{d}\Omega + Q(\mathbf{r}), \tag{2.1}$$

and we impose the boundary conditions

$$\psi(\mathbf{r}, \Omega) = f(\mathbf{r}, \Omega), \quad \text{when} \quad n(\mathbf{r}) \cdot \Omega < 0, \quad \mathbf{r} \in \partial V, \tag{2.2}$$

where $f \in L^2(\partial V, L^1(\mathbb{S}^2))$ (see Section 2.2.4). As defined in Chapter 1 the functions σ_T and σ_S are the *total* and *scattering* cross sections respectively. Together with the *absorption* cross section, σ_A, they satisfy the relation

$$\sigma_T(\mathbf{r}) = \sigma_S(\mathbf{r}) + \sigma_A(\mathbf{r}), \quad \forall \mathbf{r} \in V, \tag{2.3}$$

and are all strictly positive over the whole spatial domain. For the majority of this thesis we will assume that the cross sections are piecewise smooth and any assumptions

contrary to this will be explicitly stated.

To simplify this exposition we introduce an operator $\mathcal{T}$, called the *transport operator*, which is defined to be

$$\mathcal{T}\psi(\mathbf{r},\Omega) \equiv \Omega \cdot \nabla\psi(\mathbf{r},\Omega) + \sigma_T(\mathbf{r})\psi(\mathbf{r},\Omega). \tag{2.4}$$

With this, the problem we are interested in solving can be written as

$$\mathcal{T}\psi(\mathbf{r},\Omega) = \sigma_S(\mathbf{r})\phi(\mathbf{r}) + Q(\mathbf{r}), \tag{2.5}$$

together with the boundary conditions (2.2). Here ϕ is known as the *scalar flux*, and is defined as the neutron flux averaged over all angles, i.e.

$$\phi(\mathbf{r}) \equiv \mathcal{P}\psi(\mathbf{r},\Omega), \tag{2.6}$$

with $\mathcal{P}$ defined as

$$\mathcal{P}(\cdot) \equiv \frac{1}{4\pi}\int_{\mathbb{S}^2}(\cdot)\,\mathrm{d}\Omega. \tag{2.7}$$

The scalar flux is an important quantity which we will focus on during the next three chapters. Though it contains less information than the neutron flux, it is often the case that the direction particles are travelling in is not required (see [57, Section 1-3]). Also we will see in Section 2.4 that one step of the basic source iteration algorithm (Algorithm 1) can yield the full neutron flux provided one knows the scalar flux. For these reasons it is a central quantity of interest.

2.2.2 Transport Equation in 2D

In two dimensions the neutron flux is given by $\psi(\tilde{\mathbf{r}},\tilde{\Omega})$, where $\tilde{\mathbf{r}} \in V$ with V a bounded domain in $\mathbb{R}^2$ and $\tilde{\Omega} \in \mathbb{S}^1$. The monoenergetic, steady-state transport equation with isotropic neutron interactions and isotropic neutron source governing the neutron flux in 2D is given by

$$\tilde{\Omega}\cdot\tilde{\nabla}\psi(\tilde{\mathbf{r}},\tilde{\Omega}) + \sigma_T(\tilde{\mathbf{r}})\psi(\tilde{\mathbf{r}},\tilde{\Omega}) = \frac{\sigma_S(\tilde{\mathbf{r}})}{2\pi}\int_{\mathbb{S}^1}\psi(\tilde{\mathbf{r}},\tilde{\Omega})\,\mathrm{d}\tilde{\Omega} + Q(\tilde{\mathbf{r}}), \tag{2.8}$$

where $\tilde{\nabla}$ denotes the 2D gradient. The flux, ψ, is subject to the boundary conditions

$$\psi(\tilde{\mathbf{r}},\tilde{\Omega}) = f(\tilde{\mathbf{r}},\tilde{\Omega}), \quad \text{when} \quad \tilde{n}(\tilde{\mathbf{r}})\cdot\tilde{\Omega} < 0, \quad \tilde{\mathbf{r}} \in \partial V. \tag{2.9}$$

where $f \in L^2(\partial V, L^1(\mathbb{S}^1))$. Again, the cross sections are assumed piecewise smooth unless we state otherwise. They are all strictly positive at all points in the domain,

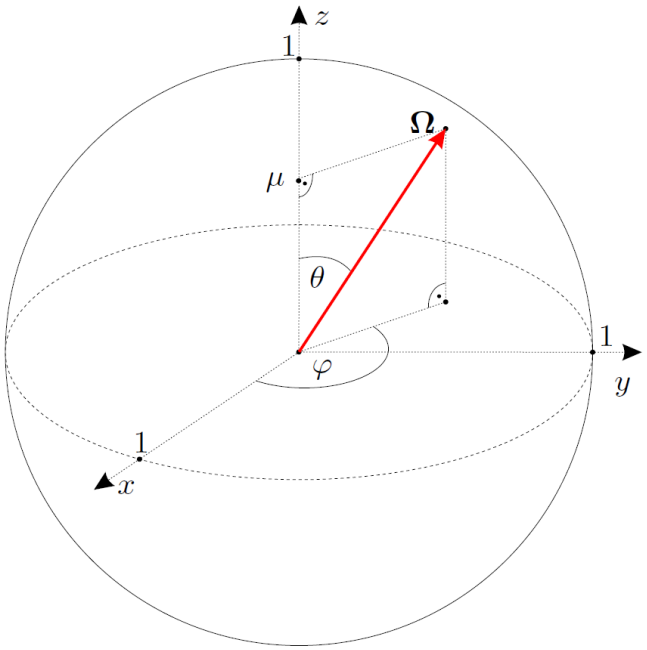

Figure 2-1: Standard polar coordinates on $\mathbb{S}^2$, showing the contribution of Ω in the z-direction. Taken from [67].

and satisfy $\sigma_T(\tilde{\mathbf{r}}) = \sigma_S(\tilde{\mathbf{r}}) + \sigma_A(\tilde{\mathbf{r}})$.

As in the 3D case, we can define a 2D version of the transport operator, $\mathcal{T}$, via

$$\mathcal{T}\psi(\tilde{\mathbf{r}}, \tilde{\Omega}) \equiv \tilde{\Omega} \cdot \nabla\psi(\tilde{\mathbf{r}}, \tilde{\Omega}) + \sigma_T(\tilde{\mathbf{r}})\psi(\tilde{\mathbf{r}}, \tilde{\Omega}). \tag{2.10}$$

So we can specify the problem of interest by

$$\mathcal{T}\psi(\tilde{\mathbf{r}}, \tilde{\Omega}) = \sigma_S(\tilde{\mathbf{r}})\phi(\tilde{\mathbf{r}}) + Q(\tilde{\mathbf{r}}), \tag{2.11}$$

together with the boundary conditions (2.9). Here the scalar flux, ϕ, is defined via a 2D version of the averaging operator $\mathcal{P}$, specifically

$$\phi(\tilde{\mathbf{r}}) = \mathcal{P}\psi(\tilde{\mathbf{r}}, \tilde{\Omega}) \equiv \frac{1}{2\pi}\int_{\mathbb{S}^1} \psi(\tilde{\mathbf{r}}, \tilde{\Omega})\, \mathrm{d}\tilde{\Omega}. \tag{2.12}$$

2.2.3 Transport Equation in 1D

In one dimension the neutron flux is a function of space, $x \in V$ with V a bounded domain in $\mathbb{R}$, and angle, $\mu \in \mathbb{S}^0 = [-1, 1]$, which is the contribution of Ω in the z-direction in 3D polar coordinates (see Figure 2-1). The variable μ then parametrises the unit 'sphere' in 1D via $\mu = \cos(\theta)$ for $\theta \in [0, \pi]$. This 1D geometry is often referred to as

slab geometry, with a very good description given in Prinja and Larsen, [59, Section 3.2]. In this thesis we will usually specify $V = [x_L, x_R]$. The monoenergetic, steady-state transport equation with isotropic neutron interactions and isotropic neutron source governing this 1D flux is given by

$$\mu \frac{\partial}{\partial x}\psi(x,\mu) + \sigma_T(x)\psi(x,\mu) = \frac{\sigma_S(x)}{2}\int_{[-1,1]} \psi(x,\mu)\ \mathrm{d}\mu + Q(x), \tag{2.13}$$

subject to the boundary conditions

$$\begin{aligned} \psi(x_L,\mu) &= f_L(\mu), \quad \text{when } \mu > 0, \\ \psi(x_R,\mu) &= f_R(\mu), \quad \text{when } \mu < 0, \end{aligned} \tag{2.14}$$

where $f_L, f_R \in L^1[-1,1]$. These are a 1D form of the *boundary source* conditions given in Section 1.1.2. As in 3D, the cross sections are all piecewise smooth, strictly positive at all points in the domain, and satisfy the property $\sigma_T(x) = \sigma_S(x) + \sigma_A(x)$ for all $x \in V$.

As in the 3D case, we can define a 1D version of the transport operator, $\mathcal{T}$, via

$$\mathcal{T}\psi(x,\mu) \equiv \mu \frac{\partial}{\partial x}\psi(x,\mu) + \sigma_T(x)\psi(x,\mu). \tag{2.15}$$

So the problem of interest can be written concisely as

$$\mathcal{T}\psi(x,\mu) = \sigma_S(x)\phi(x) + Q(x), \tag{2.16}$$

together with the boundary conditions (2.14), with the scalar flux, ϕ, defined using a 1D version of the operator $\mathcal{P}$ as

$$\phi(x) \equiv \mathcal{P}\psi(x,\mu) \equiv \frac{1}{2}\int_{[-1,1]} \psi(x,\mu)\ \mathrm{d}\mu. \tag{2.17}$$

2.2.4 Notation and Space

To allow mathematical rigour in our later analysis, we shall specify the space we are working within. In 3D, with $V \subset \mathbb{R}^3$ bounded, define

$$L^2\left(V, L^1(\mathbb{S}^2)\right) \equiv \left\{ \psi : V \times \mathbb{S}^2 \to \mathbb{R} \ :\ \int_V \|\psi(\mathbf{r},\cdot)\|^2_{L^1(\mathbb{S}^2)}\ \mathrm{d}\mathbf{r} < \infty \right\},$$

and for $\psi \in L^2\left(V, L^1(\mathbb{S}^2)\right)$ we write

$$\|\psi\|_{L^2(V,L^1(\mathbb{S}^2))} = \left(\int_V \|\psi(\mathbf{r},\cdot)\|^2_{L^1(\mathbb{S}^2)}\ \mathrm{d}\mathbf{r} \right)^{1/2}.$$

Here

$$\|\psi(\mathbf{r},\cdot)\|_{L^1(\mathbb{S}^2)} \equiv \int_{\mathbb{S}^2} |\psi(\mathbf{r},\Omega)| \, \mathrm{d}\Omega,$$

is the usual L^1-norm with respect to angle. Therefore we can say in 3D that in solving (2.5) we are looking for some $\psi \in L^2\left(V, L^1(\mathbb{S}^2)\right)$ that satisfies boundary conditions, (2.2), with $\nabla \cdot \psi \in L^2\left(V, L^1(\mathbb{S}^2)\right)$ also. The scalar flux, ϕ, will be shown to lie within the space $L^2(V)$, defined with the usual norm $\|\cdot\|_{L^2(V)}$. These definitions apply also for 2D and 1D, but using the appropriate spatial and angular domains.

We will use the operator norm as defined by

$$\|\mathcal{A}\|_{\mathscr{L}(L^2(V))} \equiv \sup_{v \in L^2(V)} \left\{ \frac{\|\mathcal{A}v\|_{L^2(V)}}{\|v\|_{L^2(V)}} : v \neq 0 \right\},$$

for any bounded linear operator $\mathcal{A} : L^2(V) \to L^2(V)$.

2.3 Solution to the Transport Equation in the Absence of Scattering

We will begin this section by deriving neutron flux and scalar flux solutions to the transport equation without scattering under non-zero boundary conditions in each dimension. Using these we will specify a solution operator, denoted $\mathcal{K}_{\sigma_T}$, for the transport equation without scattering in each dimension, but with the further restrictions of zero boundary conditions and a homogeneous right hand side. We will go on to prove various properties of this operator, and in particular we will prove that it is a *positive-definite* operator (see Lemma 2.13).

This solution operator (sometimes denoted just $\mathcal{K}$) has been considered in the literature as far back as Kaper and Kellog, 1977, [45] and Duderstadt and Martin, 1979, [24, Chapter 2], and was more recently derived in Scheben, 2011, [67, Chapter 2]. Our work is motivated by that of [67], however we are using spatially dependent cross sections.

2.3.1 Formulae for Angular and Scalar Fluxes

First we define some notation concerning distance in a 3D setting. For any neutron at position $\mathbf{r} \in V \subset \mathbb{R}^3$ travelling in direction $\Omega \in \mathbb{S}^2$, we define the position $\mathbf{r}' \equiv \mathbf{r} - s\Omega$, with $s \in \mathbb{R}^+$, thus $\mathbf{r}$ is a distance s from $\mathbf{r}'$ (see Figure 2-2 for a 2D illustration). We will denote by s^b the specific distance s such that $\mathbf{r}'$ lies in the boundary, ∂V, and we will denote this boundary point by $\mathbf{r}^b$. This can be rigorously defined as

$$\mathbf{r}^b \equiv \mathbf{r} - s^b\Omega, \tag{2.18}$$

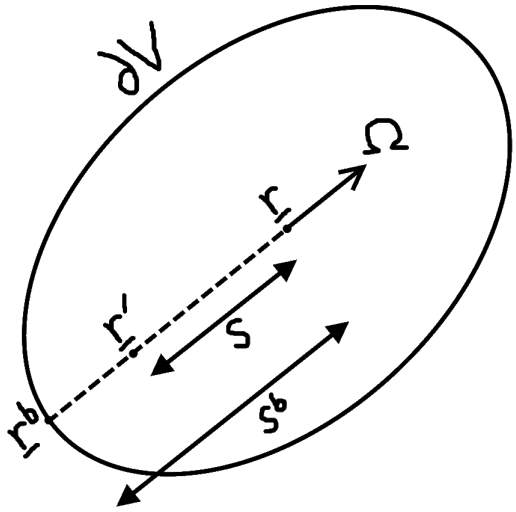

Figure 2-2: Distances related to a neutron in 2D.

with

$$s^b(\mathbf{r}, \Omega) \equiv \max\{s \geq 0 : \mathbf{r} - s\Omega \in \partial V\}.$$

We will also find the following definition useful.

Definition 2.1 (Optical Path Length)**:**
The optical path length *between* $\mathbf{r}$ *and* $\mathbf{r}'$ *is denoted* $\tau(\mathbf{r}, \mathbf{r}')$*, and is defined by*

$$\tau(\mathbf{r}, \mathbf{r}') \equiv \int_{l(\mathbf{r},\mathbf{r}')} \sigma_T(\mathbf{z})\ dl(\mathbf{z}),$$

see [12, Section 1.2b, p.24] for details. Here, $l(\mathbf{r}, \mathbf{r}')$ *is the straight line from* $\mathbf{r}$ *to* $\mathbf{r}'$*, and the notation* $dl(z)$ *represents the integral along this line. Note that in 1D,*

$$\tau(x, y) \equiv \left| \int_x^y \sigma_T(z)\ dz \right|,$$

and so $\tau(x, y) = \tau(y, x)$ *as in the higher dimensions.*

With this in hand we can prove the following result concerning the solution to the transport equation without scattering. This result is similar to [67, Lemma 2.1] however we allow for spatially dependent cross sections and non-zero boundary conditions.

Lemma 2.2:
Let $g \in L^2(V, L^\infty(\mathbb{S}^2))$ *and consider the problem of solving*

$$\mathcal{T}\psi(\mathbf{r}, \Omega) \equiv \Omega \cdot \nabla\psi(\mathbf{r}, \Omega) + \sigma_T(\mathbf{r})\psi(\mathbf{r}, \Omega) = g(\mathbf{r}, \Omega), \tag{2.19}$$

with $\mathbf{r} \in V \subset \mathbb{R}^3$ *and* $\Omega \in \mathbb{S}^2$*. Here* ψ *is subject to the boundary conditions*

$$\psi(\mathbf{r}, \Omega) = f(\mathbf{r}, \Omega), \quad \forall \mathbf{r} \in \partial V, \ \ \textit{such that } n(\mathbf{r}) \cdot \Omega < 0,$$

where $f \in L^2(\partial V, L^1(\mathbb{S}^2))$*. This problem has a unique solution*

$$\psi(\mathbf{r}, \Omega) = f(\mathbf{r}^b, \Omega) \exp\left(-\tau(\mathbf{r}, \mathbf{r}^b)\right) + \int_{l(\mathbf{r}, \mathbf{r}^b)} g(\mathbf{r}', \Omega) \exp\left(-\tau(\mathbf{r}, \mathbf{r}')\right) \, dl(\mathbf{r}'), \tag{2.20}$$

where $\mathbf{r}^b$ *is defined in* (2.18)*.*

Proof.
Throughout this proof, where it is not ambiguous, we will abbreviate $s^b(\mathbf{r}, \Omega)$ by s^b. Using this we have a point $\mathbf{r}^b \in \partial V$ associated with any pair $(\mathbf{r}, \Omega) \in V \times \mathbb{S}^2$. More generally, we can define $\mathbf{r}'(s)$ as

$$\mathbf{r}'(s) = \mathbf{r} - s\Omega,$$

with $0 \leq s \leq s^b$. Then $\|\mathbf{r} - \mathbf{r}'(s)\|_2 = s$ and $\mathbf{r}'(s^b) = \mathbf{r}^b$.

First let us evaluate (2.19) at a point $\mathbf{r}'(s)$, and then multiply both sides by the integrating factor $\exp(-\tau(\mathbf{r}, \mathbf{r}'))$. This leaves

$$\Big[\Omega \cdot \nabla \psi(\mathbf{r}'(s), \Omega) + \sigma_T(\mathbf{r}'(s)) \psi(\mathbf{r}'(s), \Omega)\Big] \exp\left(-\tau(\mathbf{r}, \mathbf{r}'(s))\right) = g(\mathbf{r}'(s), \Omega) \exp\left(-\tau(\mathbf{r}, \mathbf{r}'(s))\right).$$

This can be rewritten as

$$-\frac{\mathrm{d}}{\mathrm{d}s}\Big[\psi(\mathbf{r} - s\Omega, \Omega) \exp\left(-\tau(\mathbf{r}, \mathbf{r} - s\Omega)\right)\Big] = g(\mathbf{r}'(s), \Omega) \exp\left(-\tau(\mathbf{r}, \mathbf{r}'(s))\right). \tag{2.21}$$

We now integrate (2.21) over s from 0 to s^b. Since this amounts to integrating $\mathbf{r}'$ along the line $l(\mathbf{r}, \mathbf{r}^b)$, the right hand side of (2.21) simply becomes

$$\int_{l(\mathbf{r}, \mathbf{r}^b)} g(\mathbf{r}', \Omega) \exp\left(-\tau(\mathbf{r}, \mathbf{r}')\right) \, \mathrm{d}l(\mathbf{r}') \tag{2.22}$$

Next we focus on just the left hand side of (2.21), where integrating over s results in

$$\int_0^{s^b} -\frac{\mathrm{d}}{\mathrm{d}s}\Big[\psi(\mathbf{r}-s\Omega,\Omega)\exp\left(-\tau(\mathbf{r},\mathbf{r}-s\Omega)\right)\Big]\,\mathrm{d}s$$

$$= -\psi(\mathbf{r}-s^b\Omega,\Omega)\exp\Big(-\tau(\mathbf{r},\mathbf{r}-s^b\Omega)\Big) + \psi(\mathbf{r},\Omega)\exp\left(-\tau(\mathbf{r},\mathbf{r})\right)$$

$$= \psi(\mathbf{r},\Omega) - \psi(\mathbf{r}^b,\Omega)\exp\Big(-\tau(\mathbf{r},\mathbf{r}^b)\Big). \tag{2.23}$$

Now we know that $\mathbf{r}-s^b\Omega \in \partial V$, and also note that at this point on the boundary Ω points inwards to the domain and so $n(\mathbf{r}-s^b\Omega)\cdot\Omega < 0$. This follows since the domain is convex, and so any point on the line joining $\mathbf{r}^b$ and $\mathbf{r}$ lies in V. Therefore our boundary condition tells us that $\psi(\mathbf{r}-s^b\Omega,\Omega) = f(\mathbf{r}-s^b\Omega,\Omega)$. To conclude the proof we can combine these boundary conditions with (2.23) and (2.22) to obtain (2.20), as required.

To verify uniqueness, we just need to assume the existence of two solutions and show that their difference is always zero. This is done by taking $g=0$ and $f=0$, then solving (2.19) as above. This does lead to a zero flux, and hence the solution (2.20) is unique. □

Corollary 2.3:
The scalar flux ϕ associated with the neutron flux (2.20) *is given by*

$$\phi(\mathbf{r}) = \int_{\mathbb{S}^2} f(\mathbf{r}^b,\Omega)k_{\sigma_T}(\mathbf{r},\mathbf{r}^b)\left\|\mathbf{r}-\mathbf{r}^b\right\|_2^2\, d\Omega + \int_V g(\mathbf{r}',\Omega(\mathbf{r},\mathbf{r}'))k_{\sigma_T}(\mathbf{r},\mathbf{r}')\, d\mathbf{r}', \tag{2.24}$$

where $\Omega(\mathbf{r},\mathbf{r}') \equiv (\mathbf{r}-\mathbf{r}')/\left\|\mathbf{r}-\mathbf{r}'\right\|_2$, *and*

$$k_{\sigma_T}(\mathbf{r},\mathbf{r}') \equiv \frac{\exp\left(-\tau(\mathbf{r},\mathbf{r}')\right)}{4\pi\left\|\mathbf{r}-\mathbf{r}'\right\|_2^2}. \tag{2.25}$$

Proof.
The scalar flux is defined to be

$$\phi(\mathbf{r}) \equiv \frac{1}{4\pi}\int_{\mathbb{S}^2}\psi(\mathbf{r},\Omega)\,\mathrm{d}\Omega.$$

Integrating the boundary term in (2.20) over angle and scaling we find

$$\frac{1}{4\pi}\int_{\mathbb{S}^2} f(\mathbf{r}^b,\Omega)\exp\left(-\tau(\mathbf{r},\mathbf{r}^b)\right)\,\mathrm{d}\Omega = \int_{\mathbb{S}^2} f(\mathbf{r}^b,\Omega)k_{\sigma_T}(\mathbf{r},\mathbf{r}^b)\left\|\mathbf{r}-\mathbf{r}^b\right\|_2^2\,\mathrm{d}\Omega.$$

Next we focus on the right hand term in (2.20). We know that

$$\int_{l(\mathbf{r},\mathbf{r}')} \sigma_T(\mathbf{z})\, \mathrm{d}l(\mathbf{z}) \equiv \int_0^{\|\mathbf{r}-\mathbf{r}'\|_2} \sigma_T\left(\mathbf{r} - s\frac{(\mathbf{r}-\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|_2}\right)\, \mathrm{d}s,$$

therefore if we average over angle, we get

$$\begin{aligned}
&\frac{1}{4\pi}\int_{\mathbb{S}^2}\int_{l(\mathbf{r},\mathbf{r}^b)} g(\mathbf{r}',\Omega)\exp\left(-\tau(\mathbf{r},\mathbf{r}')\right)\, \mathrm{d}l(\mathbf{r}')\, \mathrm{d}\Omega \\
&\quad = \frac{1}{4\pi}\int_{\mathbb{S}^2}\int_0^{\|\mathbf{r}-\mathbf{r}^b\|_2} g\left(\mathbf{r} - s\frac{(\mathbf{r}-\mathbf{r}^b)}{\|\mathbf{r}-\mathbf{r}^b\|_2}, \Omega\right)\exp\left(-\tau\left(\mathbf{r}, \mathbf{r} - s\frac{(\mathbf{r}-\mathbf{r}^b)}{\|\mathbf{r}-\mathbf{r}^b\|_2}\right)\right)\, \mathrm{d}s\, \mathrm{d}\Omega.
\end{aligned} \tag{$\dagger$}$$

By definition we know $\mathbf{r}^b = \mathbf{r} - s^b\Omega$, and so

$$\frac{(\mathbf{r}-\mathbf{r}^b)}{\|\mathbf{r}-\mathbf{r}^b\|_2} = \frac{s^b\Omega}{\|s^b\Omega\|_2} = \Omega.$$

Thus

$$(\dagger) = \frac{1}{4\pi}\int_{\mathbb{S}^2}\int_0^{s^b} g\left(\mathbf{r} - s\Omega, \Omega\right)\frac{\exp\left(-\tau\left(\mathbf{r}, \mathbf{r} - s\Omega\right)\right)}{s^2} s^2\, \mathrm{d}s\, \mathrm{d}\Omega, \tag{$\ddagger$}$$

and taking spherical coordinates centred at $\mathbf{r}$ with $\mathbf{r}' = \mathbf{r} - s\Omega$, so $\|\mathbf{r}-\mathbf{r}'\|_2 = s$ then results in

$$(\ddagger) = \int_V g(\mathbf{r}', \Omega(\mathbf{r},\mathbf{r}'))\frac{\exp\left(-\tau(\mathbf{r},\mathbf{r}')\right)}{4\pi\|\mathbf{r}-\mathbf{r}'\|_2^2}\, \mathrm{d}\mathbf{r}'.$$

From here (2.24) follows immediately. □

Equivalent results to Lemma 2.2 can be proved in 2D and 1D also, however the argument in Corollary 2.3 will only extend to 2D, with 1D requiring us to further restrict $g \in L^2[x_L, x_R]$. We will give the statements of these extensions informally here. Firstly in 2D we consider the transport equation without scattering, given by

$$\tilde{\Omega}\cdot\nabla\psi(\tilde{\mathbf{r}},\tilde{\Omega}) + \sigma_T(\tilde{\mathbf{r}})\psi(\tilde{\mathbf{r}},\tilde{\Omega}) = g(\tilde{\mathbf{r}},\tilde{\Omega}),$$

subject to boundary conditions (2.9), where $\tilde{\mathbf{r}} \in V \subset \mathbb{R}^2$, $\tilde{\Omega} \in \mathbb{S}^1$ and we choose $g \in L^2\left(V, L^\infty(\mathbb{S}^1)\right)$. The solution is given by

$$\psi(\tilde{\mathbf{r}},\tilde{\Omega}) = f(\tilde{\mathbf{r}}^b,\tilde{\Omega})\exp\left(-\tau(\tilde{\mathbf{r}},\tilde{\mathbf{r}}^b)\right) + \int_{l(\tilde{\mathbf{r}},\tilde{\mathbf{r}}^b)} g(\tilde{\mathbf{r}}',\tilde{\Omega})\exp\left(-\tau(\tilde{\mathbf{r}},\tilde{\mathbf{r}}')\right)\,\mathrm{d}l(\tilde{\mathbf{r}}') \tag{2.26}$$

Next applying the 2D version of the operator $\mathcal{P}$ to (2.26) we can obtain

$$\phi(\mathbf{r}) = \int_{\mathbb{S}^1} f(\tilde{\mathbf{r}}^b,\tilde{\Omega})k_{\sigma_T}(\tilde{\mathbf{r}},\tilde{\mathbf{r}}^b)\left\|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}^b\right\|_2 \,\mathrm{d}\tilde{\Omega} + \int_V g(\tilde{\mathbf{r}}',\tilde{\Omega}(\tilde{\mathbf{r}},\tilde{\mathbf{r}}'))k_{\sigma_T}(\tilde{\mathbf{r}},\tilde{\mathbf{r}}')\,\mathrm{d}\tilde{\mathbf{r}}', \tag{2.27}$$

where $\tilde{\Omega}(\tilde{\mathbf{r}},\tilde{\mathbf{r}}') \equiv (\tilde{\mathbf{r}}-\tilde{\mathbf{r}}')/\left\|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'\right\|_2$, and

$$k_{\sigma_T}(\tilde{\mathbf{r}},\tilde{\mathbf{r}}') \equiv \frac{\exp\left(-\tau(\tilde{\mathbf{r}},\tilde{\mathbf{r}}')\right)}{2\pi\left\|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'\right\|_2}.$$

In 1D, we want the solution to

$$\mu\frac{\partial}{\partial x}\psi(x,\mu) + \sigma_T(x)\psi(x,\mu) = g(x,\mu)$$

subject to the boundary conditions (2.14) where $x \in [x_L,x_R]$, $\mu \in [-1,1]$, and we choose $g \in L^2([x_L,x_R],L^\infty[-1,1])$. The solution is given by

$$\psi(x,\mu) = \begin{cases} f_L(\mu)e^{\frac{-1}{\mu}\tau(x_L,x)} + \dfrac{1}{\mu}\displaystyle\int_{x_L}^{x} g(x',\mu)e^{\frac{-1}{\mu}\int_{x'}^{x}\sigma_T(z)\,\mathrm{d}z}\,\mathrm{d}x', & \mu > 0, \\ f_R(\mu)e^{\frac{1}{\mu}\tau(x,x_R)} - \dfrac{1}{\mu}\displaystyle\int_{x}^{x_R} g(x',\mu)e^{\frac{1}{\mu}\int_{x}^{x'}\sigma_T(z)\,\mathrm{d}z}\,\mathrm{d}x', & \mu < 0. \end{cases} \tag{2.28}$$

Remark 2.4:
The behaviour of the solution (2.28) *for* μ *approaching zero is non-trivial. It was subject to analysis by Kaper and Kellogg [45] where the differentiability and continuity of* (2.28) *are considered. A more general consideration of integral operators of this form is provided in [31].*

As mentioned above, before applying the 1D version of the operator $\mathcal{P}$ we need to assume that the right hand side is only spatially dependent, i.e. $g \in L^2[x_L,x_R]$. Under this assumption we can apply $\mathcal{P}$ to (2.28) and find

$$\phi(x) = \frac{1}{2}\int_0^1 f_L(\mu)e^{\frac{-1}{\mu}\tau(x_L,x)}\,\mathrm{d}\mu + \frac{1}{2}\int_{-1}^0 f_R(\mu)e^{\frac{1}{\mu}\tau(x,x_R)}\,\mathrm{d}\mu + \int_{x_L}^{x_R} g(x')k_{\sigma_T}(x,x')\,\mathrm{d}x' \tag{2.29}$$

where

$$k_{\sigma_T}(x,x') \equiv \frac{1}{2}\int_0^1 \frac{1}{\mu}\exp\left(\frac{-1}{\mu}\tau(x,x')\right)\,\mathrm{d}\mu,$$
$$\equiv \frac{1}{2}E_1(\tau(x,x')). \tag{2.30}$$

Here E_1 denotes the exponential integral function, defined in Abramowitz and Stegun, [1, Equation 5.1.4], as

$$E_1(z) \equiv \int_1^\infty \frac{\exp(-zt)}{t}\,\mathrm{d}t.$$

If the total cross section, σ_T, is constant then $\tau(x,x') = \sigma_T|x-x'|$ and (2.30) reduces to the 1D kernel studied in F. Scheben 2011 [67] (see equation (2.16)).

2.3.2 Zero Boundary Conditions: the operator $\mathcal{K}_{\sigma_T}$

Corollary 2.3 and its lower dimensional counterparts ((2.27) and (2.29)) enable us to give an integral form of a solution operator, $\mathcal{K}_{\sigma_T}$, for the transport equation without scattering. In each dimension we now assume that the boundary function $f = 0$, and we work in the special case $g \in L^2(V)$. We consider each dimension in turn. Firstly in 3D we consider Corollary 2.3 under these assumptions and find

$$\phi(\mathbf{r}) = (\mathcal{K}_{\sigma_T}g)(\mathbf{r}) \equiv \int_V g\left(\mathbf{r}'\right) k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}', \tag{2.31}$$

where $\mathbf{r} \in V \subset \mathbb{R}^3$, and

$$k_{\sigma_T}(\mathbf{r},\mathbf{r}') \equiv \frac{\exp(-\tau(\mathbf{r},\mathbf{r}'))}{4\pi\left\|\mathbf{r}-\mathbf{r}'\right\|_2^2}. \tag{2.32}$$

Similarly in 2D we use (2.27) and working under our assumptions we obtain

$$\phi(\tilde{\mathbf{r}}) = (\mathcal{K}_{\sigma_T}g)(\tilde{\mathbf{r}}) \equiv \int_V g\left(\tilde{\mathbf{r}}'\right) k_{\sigma_T}(\tilde{\mathbf{r}},\tilde{\mathbf{r}}')\,\mathrm{d}\tilde{\mathbf{r}}', \tag{2.33}$$

where now $\tilde{\mathbf{r}} \in V \subset \mathbb{R}^2$, and

$$k_{\sigma_T}(\tilde{\mathbf{r}},\tilde{\mathbf{r}}') \equiv \frac{\exp(-\tau(\tilde{\mathbf{r}},\tilde{\mathbf{r}}'))}{2\pi\left\|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'\right\|_2}. \tag{2.34}$$

Lastly in 1D we work from (2.29) and simply set the boundary conditions to zero leaving behind

$$\phi(x) = (\mathcal{K}_{\sigma_T}g)(x) \equiv \int_{x_L}^{x_R} g\left(x'\right) k_{\sigma_T}(x,x')\,\mathrm{d}x', \tag{2.35}$$

where $x \in [x_L, x_R] \subset \mathbb{R}$, and

$$k_{\sigma_T}(x, x') \equiv \frac{1}{2} E_1(\tau(x, x')). \tag{2.36}$$

2.3.3 Properties of the operator, $\mathcal{K}_{\sigma_T}$

In this section we aim to establish that the operator $\mathcal{K}_{\sigma_T}$ is a *positive-definite*, *compact* operator. We will first establish the compactness of $\mathcal{K}_{\sigma_T}$ in 2D and 3D in Lemma 2.6, and in 1D in Lemma 2.8. Next we will show (in all dimensions) that $\mathcal{K}_{\sigma_T}$ is self-adjoint in Lemma 2.11 before concluding that it is a positive-definite operator in Lemma 2.13.

So we start by proving compactness of the operator, and focus first on the 2 and 3 dimensional cases. We will require the following result from [44, Theorem 6, p.332], that establishes compactness for operators that fulfil certain requirements.

Lemma 2.5:
Consider $L^2(D)$, where D is a d-dimensional bounded region in Euclidean space, and define the kernel of potential type

$$k(\mathbf{r}, \mathbf{r}') = \frac{b(\mathbf{r}, \mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|_2^m},$$

where $b(\mathbf{r}, \mathbf{r}')$ is a bounded function, continuous for $\mathbf{r} \neq \mathbf{r}'$. If $d > m$ then the integral operator

$$\left(\mathcal{K}_{\sigma_T} g\right)(\mathbf{r}) = \int_D k(\mathbf{r}, \mathbf{r}') g(\mathbf{r}') \, d\mathbf{r}', \quad \mathbf{r} \in D,$$

is a compact operator mapping $L^2(D)$ into $L^2(D)$.

Using Lemma 2.5 we can show the compactness of $\mathcal{K}_{\sigma_T}$ in 2 and 3 dimensions

Lemma 2.6:
$\mathcal{K}_{\sigma_T}$ is a compact operator in 2 and 3 dimensions on $L^2(V)$.

Proof.
We work first in 3D. In order to utilise Lemma 2.5 we set

$$b(\mathbf{r}, \mathbf{r}') \equiv \frac{1}{4\pi} \exp\left(-\tau(\mathbf{r}, \mathbf{r}')\right).$$

Then, with $d = 3$, $m = 2$ and $D = V \subset \mathbb{R}^3$, our kernel $k_{\sigma_T} = b(\mathbf{r}, \mathbf{r}') / \|\mathbf{r} - \mathbf{r}'\|_2^2$ is a *kernel of potential type* satisfying the assumptions of Lemma 2.5, and thus $\mathcal{K}_{\sigma_T}$ is compact. A similar argument holds in the 2 dimensional case using the 2D kernel (2.34). □

In 1D our kernel, k_{σ_T}, is not of potential type and so to establish compactness we require the following result from [38, Theorem 7, p.51].

Lemma 2.7:
Let $k(x,y)$ be such that

$$\int_{x_L}^{x_R}\int_{x_L}^{x_R} |k(x,y)|^2 \; dx \; dy < \infty. \tag{2.37}$$

Then the operator

$$\left(\mathcal{K}_{\sigma_T} g\right)(x) = \int_{x_L}^{x_R} k(x,y) g(y) \; dy$$

is a compact operator on $L^2([x_L, x_R])$.

A kernel that is square integrable, satisfying (2.37), is known as a *Hilbert-Schmidt kernel* (see [38]).

Lemma 2.8:
$\mathcal{K}_{\sigma_T}$ is a compact operator in $L^2([x_L, x_R])$.

Proof.
To show that $\mathcal{K}_{\sigma_T}$ is compact in 1D, Lemma 2.7 requires us to demonstrate that

$$\int_{x_L}^{x_R}\int_{x_L}^{x_R} |k_{\sigma_T}(x,y)|^2 \; \mathrm{d}x \; \mathrm{d}y < \infty,$$

where in our case

$$\begin{aligned} k_{\sigma_T}(x,y) &= \frac{1}{2} E_1\left(\tau(x,y)\right) \\ &= \frac{1}{2}\int_0^1 \frac{1}{\mu} e^{\frac{-1}{\mu}\tau(x,y)} \; \mathrm{d}\mu \end{aligned}$$

Applying the change of variable $\eta = \mu / |x-y|$ for $|x-y| \in (0, x_R - x_L)$, the kernel becomes

$$\begin{aligned} k_{\sigma_T}(x,y) &= \frac{1}{2}\int_0^{\frac{1}{|x-y|}} \frac{1}{\eta} e^{\frac{-1}{\eta|x-y|}\tau(x,y)} \; \mathrm{d}\eta \\ &= \frac{1}{2}\underbrace{\int_0^1 \frac{1}{\eta} e^{\frac{-1}{\eta|x-y|}\tau(x,y)} \; \mathrm{d}\eta}_{(I_1)} + \frac{1}{2}\underbrace{\int_1^{\frac{1}{|x-y|}} \frac{1}{\eta} e^{\frac{-1}{\eta|x-y|}\tau(x,y)} \; \mathrm{d}\eta}_{(I_2)}. \end{aligned}$$

Next, we note that

$$\begin{aligned}(I_1) &= \int_0^1 \frac{1}{\eta} e^{\frac{-1}{\eta|x-y|}\int_x^y \sigma_T(t)\,\mathrm{d}t}\,\mathrm{d}\eta \\ &\leq \int_0^1 \frac{1}{\eta} e^{\frac{-1}{\eta}\min\limits_{(y,x)}|\sigma_T|\int_x^y \frac{1}{|x-y|}\,\mathrm{d}t}\,\mathrm{d}\eta \\ &= \int_0^1 \frac{1}{\eta} e^{\frac{-1}{\eta}\min\limits_{(y,x)}|\sigma_T|}\,\mathrm{d}\eta \\ &= C_1, \end{aligned} \tag{2.38}$$

where C_1 is some constant. To bound (I_2) we note that

$$\exp\left(\frac{-1}{\eta\,|x-y|}\tau(x,y)\right) \leq 1,$$

since $\eta > 0$. Therefore

$$(I_2) \leq \int_1^{\frac{1}{|x-y|}} \frac{1}{\eta}\,\mathrm{d}\eta = \ln\frac{1}{|x-y|}, \tag{2.39}$$

and hence using (2.38) and (2.39) we find that

$$k_{\sigma_T}(x,y) \leq C_1 + \ln\frac{1}{|x-y|}.$$

From this point, we can follow the logic from [67, Lemma 2.14, p32] and note that

$$\ln\frac{1}{|x|} \leq \begin{cases} C_2 x^{-\frac{1}{4}}\,, & \text{if } x \in (0, e^{-4}), \\ 4\,, & \text{if } x \in (e^{-4}, x_R - x_L), \end{cases}$$

where $C_2 = 4/e$. Consequently

$$k_{\sigma_T}(x,y) \leq \begin{cases} C_1 + C_2\,|x-y|^{-\frac{1}{4}}\,, & \text{if } |x-y| \in (0, e^{-4}), \\ C_1 + 4\,, & \text{if } |x-y| \in (e^{-4}, x_R - x_L), \end{cases}$$

and so

$$\int_{x_L}^{x_R}\int_{x_L}^{x_R} |k_{\sigma_T}(x,y)|^2\,\mathrm{d}x\,\mathrm{d}y \leq \int_{x_L}^{x_R}\int_{x_L}^{x_R} \left|C_1 + C_2\,|x-y|^{-\frac{1}{4}} + 4\right|^2\,\mathrm{d}x\,\mathrm{d}y < \infty.$$

Therefore we have satisfied the requirements of Lemma 2.7, and so $\mathcal{K}_{\sigma_T}$ is a compact operator on $L^2[x_L, x_R]$. □

Remark 2.9:
We note here that having established compactness of $\mathcal{K}_{\sigma_T}$ in all dimensions, we also know that it is a bounded, linear and continuous operator. These properties will be required when showing that $\mathcal{K}_{\sigma_T}$ is self-adjoint in Lemma 2.11.

We are now in a position to establish the self-adjointness of $\mathcal{K}_{\sigma_T}$ in 1, 2 and 3 dimensions. This property is defined as follows.

Definition 2.10 (Self-Adjoint)**:**
Let H be a Hilbert space with associated inner-product $\langle \cdot, \cdot \rangle$, then a bounded linear operator $\mathcal{A} : H \to H$ is called self-adjoint *if $\langle f, \mathcal{A}g \rangle_H = \langle \mathcal{A}f, g \rangle_H$ for all $f, g \in H$.*

Lemma 2.11:
The operator $\mathcal{K}_{\sigma_T} : L^2(V) \to L^2(V)$ is a self-adjoint operator on the Hilbert space $L^2(V)$, where $V \subset \mathbb{R}^p$ for $p = 1, 2, 3$.

Proof.
In 1, 2 and 3 dimensions the kernel k_{σ_T} is symmetric. Using this in 3D, we have that

$$\begin{aligned}
\langle f, \mathcal{K}_{\sigma_T} g \rangle_{L^2(V)} &= \int_V \int_V k_{\sigma_T}(\mathbf{r}, \mathbf{r}') g(\mathbf{r}') \, \mathrm{d}\mathbf{r}' f(\mathbf{r}) \, \mathrm{d}\mathbf{r} \\
&= \int_V \int_V k_{\sigma_T}(\mathbf{r}', \mathbf{r}) f(\mathbf{r}) \, \mathrm{d}\mathbf{r} g(\mathbf{r}') \, \mathrm{d}\mathbf{r}' \\
&= \langle \mathcal{K}_{\sigma_T} f, g \rangle_{L^2(V)}.
\end{aligned}$$

We can construct equivalent arguments in 1D and 2D also, and so the proof is complete. □

Knowing that $\mathcal{K}_{\sigma_T}$ is a compact, self-adjoint operator in 1, 2 and 3 spatial dimensions, we are in a position to prove that it is positive-definite. First we will define what we mean by this.

Definition 2.12 (Positive-Definite)**:**
Let H be a Hilbert space. The operator $\mathcal{A}$ is positive-definite *if it is bounded, self-adjoint and $\langle f, \mathcal{A}f \rangle_H \geq 0$ for all $f \in H$, and $\langle f, \mathcal{A}f \rangle_H = 0$ iff $f = 0$.*

To prove that $\mathcal{K}_{\sigma_T}$ is positive-definite we turn to [67, Lemma 2.17], however we restate this result in the case of a non-constant total cross section, σ_T.

Lemma 2.13:
$\mathcal{K}_{\sigma_T}$ is a positive-definite operator on $L^2(V)$, where V is an n-dimensional closed subset of $\mathbb{R}^n$, $n = 1, 2, 3$.

Proof.
The proof follows by applying the logic of [67, Lemma 2.17] under the assumption that σ_T is piecewise smooth. □

2.4 Source Iteration Algorithm

The purpose of this section is to describe and define the basic iterative method, often called *source iteration*, that will be the focus of our convergence analysis in the next section. This method can be used to solve the neutron transport equation, and is a simple Richardson iteration (originally introduced in [62]) applied to the transport equation for the scalar flux. For the purposes of our analysis, we will present it as a two-step method.

While very basic, a discretised version of source iteration is used in modern industrial modelling codes and is also a part of many more complicated iterative methods such as diffusion synthetic acceleration (see Chapter 3) and Monte Carlo synthetic acceleration, [26].

In practice when used to solve problems where particles are likely to experience only a small number of collisions, source iteration converges quickly [2]. This situation is typically presented when either the absorption cross section is large for most of the domain, or when the system is *leaky*, i.e. particles frequently leave the domain after a small number of collisions. On the other hand, for problems in which particles usually undergo a large number of collisions, source iteration converges slowly. Typically this occurs when the chance of leakage is small and the scattering cross section is dominant over most of the domain [2]. Such a situation is referred to as *diffusive* and is very physically relevant (see for example [59, Section 8]). These diffusive regimes will be our focus in Chapter 3.

In the next section we will analyse the convergence of the continuous form of source iteration, however later we will look at numerical results obtained using a discretised form. Consequently the theoretical convergence behaviour resulting from our analysis does not directly apply. However if we succesively refine our discretisation, we expect the discrete algorithm to more and more closely approximate the continuous version. Therefore it is reasonable to expect the predicted continuous convergence behaviour to emerge.

We will now conclude this section by stating the source iteration algorithm, before moving immediately on to the convergence analysis. In Section 2.2.1 we saw that, in a 3D setting, we are trying to solve the transport equation as given by

$$\mathcal{T}\psi(\mathbf{r},\Omega) = \sigma_S(\mathbf{r})\phi(\mathbf{r}) + Q(\mathbf{r}), \tag{2.40}$$

subject to the boundary conditions

$$\psi(\mathbf{r},\Omega) = f(\mathbf{r},\Omega), \quad \text{when} \quad n(\mathbf{r})\cdot\Omega < 0, \quad \mathbf{r}\in\partial V,$$

where $f \in L^2(\partial V, L^1(\mathbb{S}^2))$. To solve this problem, source iteration is defined as follows.

Algorithm 1: Source Iteration

1. Start with some initial $\phi^{(0)}(\mathbf{r})$.
2. Solve
$$\mathcal{T}\psi^{(k+1)}(\mathbf{r},\Omega) = \sigma_S(\mathbf{r})\phi^{(k)}(\mathbf{r}) + Q(\mathbf{r}), \tag{2.41}$$
for $\psi^{(k+1)}(\mathbf{r},\Omega)$, where $\psi^{(k+1)}$ satisfies
$$\psi^{(k+1)}(\mathbf{r},\Omega) = f(\mathbf{r},\Omega), \quad \text{when} \quad n(\mathbf{r})\cdot\Omega < 0, \quad \mathbf{r}\in\partial V,$$
for some $f \in L^2(\partial V, L^1(\mathbb{S}^2))$.
3. Average over angle to find
$$\phi^{(k+1)}(\mathbf{r}) = \mathcal{P}\psi^{(k+1)}(\mathbf{r},\Omega) \equiv \frac{1}{4\pi}\int_{\mathbb{S}^2}\psi^{(k+1)}(\mathbf{r},\Omega)\,\mathrm{d}\Omega,$$
and return to step 2.

Though we have expressed a 3D version of the algorithm here, it is easily written down in a 2D or 1D setting, and our analysis in the next section will hold in all dimensions.

2.5 Convergence of Source Iteration

It is known that source iteration converges when solving the neutron transport equation with constant cross sections (see [67, Chapter 4]). Ashby et. al [7, Section 4] prove a similar result with spatially dependent cross sections for a special discrete case. This work motivated us to consider a proof in the underlying continuous case, and in Theorem 2.21 we will prove that continuous source iteration (Algorithm 1) converges

when the cross sections are piecewise smooth in space. This argument will require some preliminary results which we begin by proving in Section 2.5.1.

After our convergence work for piecewise smooth cross sections, we will turn our attention to the constant cross section case. In Section 2.5.3 we will prove new, tighter bounds on the norm of the solution operator $\mathcal{K}_{\sigma_T}$ in Theorem 2.23 for 2D and 3D, and Theorem 2.25 for 1D. These bounds will allow us to prove a new convergence result for source iteration with constant cross sections, which provides a tighter bound on the rate of convergence.

2.5.1 Preliminary Results

Throughout this section we will use V to denote any bounded subset of $\mathbb{R}^d$, with $d = 1, 2$ or 3, and the results proved will apply to all three of these dimensions. If any result needs a significantly different argument for a given dimension, this will be dealt with in the proof. We will further assume that the cross sections (σ_T, σ_S and σ_A) are piecewise smooth functions in space. During this work we will use work from Riesz and Nagy, [64], Hochstadt, [38], and Kantorovich and Akilov, [44], and will cite the appropriate results when they are used.

Our main goal in this section will be to prove the following norm bound.

Theorem 2.14:

$$\left\| \sigma_T^{1/2}(\mathbf{r}) \mathcal{K}_{\sigma_T} \sigma_T^{1/2}(\mathbf{r}) \right\|_{\mathscr{L}(L^2(V))} \leq 1.$$

Remark 2.15:
We note that this result is a natural generalisation of a result in Scheben [67] since for a constant total cross section our result reduces to $\sigma_T \left\| \mathcal{K}_{\sigma_T} \right\|_{\mathscr{L}(L^2(V))} \leq 1$*, which is precisely Theorem 2.9 in [67].*

This result will be instrumental in the proof of our main convergence result, Theorem 2.21 (see specifically (2.55)). This will be our motivation for what follows and the proof of Theorem 2.14 will be given at the end of the section. However before that we will prove results about the properties of the operator $\sigma_T^{1/2}(\mathbf{r}) \mathcal{K}_{\sigma_T} \sigma_T^{1/2}(\mathbf{r})$.

Lemma 2.16:
The operator

$$\sigma_T^{1/2}(\mathbf{r}) \mathcal{K}_{\sigma_T} \sigma_T^{1/2}(\mathbf{r}) : L^2(V) \to L^2(V)$$

is positive-definite.

Proof.
In Lemma 2.13 we saw that $\mathcal{K}_{\sigma_T}$ was positive-definite in 1D, 2D and 3D. Let $f, g \in L^2(V)$, then

$$\begin{aligned}\langle \sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2} f, g\rangle_{L^2(V)} &= \langle \mathcal{K}_{\sigma_T}\sigma_T^{1/2} f, \sigma_T^{1/2} g\rangle_{L^2(V)} \\ &= \langle \sigma_T^{1/2} f, \mathcal{K}_{\sigma_T}\sigma_T^{1/2} g\rangle_{L^2(V)} = \langle f, \sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2} g\rangle_{L^2(V)}\end{aligned}$$

thus $\sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2}$ is self-adjoint in $L^2(V)$. Next, for all $f \in L^2(V)$,

$$\langle f, \sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2} f\rangle_{L^2(V)} = \langle \sigma_T^{1/2} f, \mathcal{K}_{\sigma_T}\sigma_T^{1/2} f\rangle_{L^2(V)} \geq 0,$$

since $\mathcal{K}_{\sigma_T}$ is positive-definite. Lastly, using this we also know

$$\langle f, \sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2} f\rangle_{L^2(V)} = 0 \quad \Leftrightarrow \quad \sigma_T^{1/2} f = 0, \quad \Leftrightarrow \quad f = 0,$$

since $\sigma_T^{1/2}(\mathbf{r}) > 0$ for all $\mathbf{r} \in V$. Therefore $\sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2}$ is positive-definite as required. □

Our next result concerns a bound on the Rayleigh quotient of the operator $\sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2}$.

Lemma 2.17:
Suppose $\mathcal{L} \equiv \sigma_T^{1/2}\mathcal{K}_{\sigma_T}\sigma_T^{1/2}$, *so* $\mathcal{L} : L^2(V) \to L^2(V)$. *Then*

$$\frac{\langle g, \mathcal{L}g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \leq 1, \quad \forall g \in L^2(V)\backslash\{0\},$$

Proof.
Take any $g \in L^2(V)$ and consider

$$\mathcal{T}\psi \equiv \Omega \cdot \nabla\psi + \sigma_T\psi = \sigma_T^{1/2} g \tag{2.42}$$

where ψ satisfies zero incoming boundary conditions, (2.2) where $f = 0$. Then

$$\phi = \mathcal{K}_{\sigma_T}(\sigma_T^{1/2} g). \tag{2.43}$$

On the other hand, applying $\mathcal{P}$ directly to (2.42) we get

$$\mathcal{P}(\Omega \cdot \nabla\psi) + \sigma_T\phi = \sigma_T^{1/2} g.$$

(Note that $\sigma_T^{1/2} g$ is independent of angle, so $\mathcal{P}(\sigma_T^{1/2} g) = \sigma_T^{1/2} g$). If we multiply this through by ${\sigma_T}^{-1/2}$ and rearrange we get

$$\sigma_T^{1/2}\phi = g - {\sigma_T}^{-1/2}\mathcal{P}(\Omega\cdot\nabla\psi). \tag{2.44}$$

Combining (2.43) and (2.44) we have that

$$\mathcal{L}g = g - {\sigma_T}^{-1/2}\mathcal{P}(\Omega\cdot\nabla\psi).$$

Multiplying by g and integrating over V yields

$$\langle g, \mathcal{L}g\rangle_{L^2(V)} = \langle g, g\rangle_{L^2(V)} - \int_V {\sigma_T}^{-1/2} g\mathcal{P}(\Omega\cdot\nabla\psi)\,\mathrm{d}\mathbf{r}.$$

so

$$\frac{\langle g, \mathcal{L}g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} = 1 - \frac{1}{\langle g, g\rangle_{L^2(V)}}\int_V {\sigma_T}^{-1/2} g\mathcal{P}(\Omega\cdot\nabla\psi)\,\mathrm{d}\mathbf{r}, \tag{2.45}$$

for all $g \in L^2(V)\backslash\{0\}$. To finish we now show that the second term on the right-hand side of (2.45) is non-positive.

To begin with, note that $\langle g, g\rangle_{L^2(V)} > 0$ for all $g \in L^2(V)\backslash\{0\}$, and so we can consider the right-hand integral term without this ratio included. Set

$$\begin{aligned} A &\equiv \int_V {\sigma_T}^{-1/2} g\mathcal{P}(\Omega\cdot\nabla\psi)\,\mathrm{d}\mathbf{r} \\ &= \mathcal{P}\left(\int_V {\sigma_T}^{-1/2} g\Omega\cdot\nabla\psi\,\mathrm{d}\mathbf{r}\right). \end{aligned} \tag{2.46}$$

Then using (2.42) we have ${\sigma_T}^{-1/2} g = \psi + \sigma_T^{-1}\Omega\cdot\nabla\psi$, so

$$A = \underbrace{\mathcal{P}\left(\int_V \psi\Omega\cdot\nabla\psi\,\mathrm{d}\mathbf{r}\right)}_{\equiv A_1} + \underbrace{\mathcal{P}\left(\int_V \sigma_T^{-1}\left(\Omega\cdot\nabla\psi\right)^2\,\mathrm{d}\mathbf{r}\right)}_{\equiv A_2}.$$

We consider the two parts of this in turn. Firstly,

$$A_2 \geq 0, \tag{2.47}$$

since clearly $(\Omega\cdot\nabla\psi)^2 \geq 0$, and $\sigma_T^{-1}(\mathbf{r}) > 0$ by its definition.

Secondly we turn to A_1. Up until this point our argument has applied in 1D, 2D and 3D, with mostly notational changes needed to move between them. However we will now argue for the 2D and 3D cases only, and will look at the 1D case afterwards.

We know by the chain rule that

$$\nabla\left(\frac{1}{2}\psi^2\right) = \psi\nabla\psi,$$

and so

$$A_1 = \mathcal{P}\left(\int_V \Omega\cdot(\psi\nabla\psi)\ \mathrm{d}\mathbf{r}\right) = \mathcal{P}\left(\int_V \Omega\cdot\nabla\left(\frac{1}{2}\psi^2\right)\ \mathrm{d}\mathbf{r}\right) = \mathcal{P}\left(\int_V \nabla\cdot\left(\Omega\frac{1}{2}\psi^2\right)\ \mathrm{d}\mathbf{r}\right).$$

Then using the divergence theorem

$$A_1 = \mathcal{P}\left(\int_{\partial V}\left(\Omega\frac{1}{2}\psi^2\right)\cdot n(\mathbf{r})\ \mathrm{d}\mathbf{r}\right) = \mathcal{P}\left(\int_{\partial V}\frac{1}{2}\psi^2\left(\Omega\cdot n(\mathbf{r})\right)\ \mathrm{d}\mathbf{r}\right).$$

Now ψ^2 is always positive, and by our boundary conditions $\psi = 0$ whenever $\Omega\cdot n(\mathbf{r}) < 0$. Thus we have

$$A_1 \geq 0. \tag{2.48}$$

Next, in 1D A_1 is defined as

$$A_1 \equiv \mathcal{P}\left(\int_{[0,1]}\psi\mu\frac{\partial}{\partial x}\psi\ \mathrm{d}x\right).$$

We know by the chain rule that

$$\frac{\partial}{\partial x}\left(\frac{1}{2}\psi^2\right) = \psi\frac{\partial}{\partial x}\psi,$$

and so A_1 becomes

$$A_1 = \mathcal{P}\left(\int_{[0,1]}\frac{\partial}{\partial x}\left(\frac{\mu}{2}\psi^2\right)\ \mathrm{d}x\right),$$

$$= \mathcal{P}\left(\left[\frac{\mu}{2}\psi^2\right]_0^1\right).$$

Writing this out in full we have found that

$$A_1 = \frac{1}{4}\int_{[-1,1]}\mu\psi(1,\mu)^2\ \mathrm{d}\mu - \frac{1}{4}\int_{[-1,1]}\mu\psi(0,\mu)^2\ \mathrm{d}\mu.$$

From the boundary conditions, (2.14) with $f_L = f_R = 0$, this becomes

$$A_1 = \frac{1}{4}\int_{[0,1]} \mu\psi(1,\mu)^2 \, \mathrm{d}\mu - \frac{1}{4}\int_{[-1,0]} \mu\psi(0,\mu)^2 \, \mathrm{d}\mu \geq 0, \tag{2.49}$$

as we wanted.

From (2.47), and combining (2.48) and (2.49), we therefore know that $A \geq 0$, and so via (2.45) we have the required result. □

Remark 2.18:
In this proof we considered the problem (2.42) *together with zero incoming boundary conditions. Despite this apparent restriction, our eventual convergence result will hold for non-zero boundary conditions also. This is because later we will apply these results to an error equation rather than the transport equation itself. Therefore any boundary conditions of the form* (2.2) *applied to the transport equation result in zero boundary conditions for the error equation.*

The last result needed before we can complete the proof of Theorem 2.14 is Lemma 2.20. To prove this we will require the following result.

Theorem 2.19 ([64], Chapter 104)**:**
Every positive-definite self-adjoint operator $\mathcal{A}$ possesses a unique positive-definite self-adjoint square root, which we denote $\mathcal{A}^{1/2}$.

Lemma 2.20:
If $\mathcal{L} : L^2(V) \to L^2(V)$ is a positive-definite self-adjoint operator, then

$$\frac{\langle g, \mathcal{L}g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \leq 1 \tag{2.50}$$

implies

$$\frac{\langle g, \mathcal{L}^2 g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \leq 1, \quad \forall g \in L^2(V)\backslash\{0\}.$$

Proof.
By Theorem 2.19, $\mathcal{L}$ has a positive-definite self-adjoint square root, which we will denote $\mathcal{L}^{1/2}$.

Now let us assume (2.50) and take $g = \mathcal{L}^{1/2} f$ for some $f \in L^2(V)\backslash\{0\}$. Then using the self-adjointness of $\mathcal{L}^{1/2}$ we find

$$1 \geq \frac{\langle g, \mathcal{L}g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} = \frac{\langle \mathcal{L}^{1/2}f, \mathcal{L}\mathcal{L}^{1/2}f\rangle_{L^2(V)}}{\langle \mathcal{L}^{1/2}f, \mathcal{L}^{1/2}f\rangle_{L^2(V)}}$$
$$= \frac{\langle f, \mathcal{L}^{1/2}\mathcal{L}\mathcal{L}^{1/2}f\rangle_{L^2(V)}}{\langle f, \mathcal{L}^{1/2}\mathcal{L}^{1/2}f\rangle_{L^2(V)}} = \frac{\langle f, \mathcal{L}^2 f\rangle_{L^2(V)}}{\langle f, \mathcal{L}f\rangle_{L^2(V)}}.$$

Thus for all $f \in L^2(V)$

$$\langle f, \mathcal{L}^2 f\rangle_{L^2(V)} \leq \langle f, \mathcal{L}f\rangle_{L^2(V)},$$

and hence

$$\frac{\langle g, \mathcal{L}^2 g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \leq \frac{\langle g, \mathcal{L}g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \leq 1, \quad \forall g \in L^2(V)\backslash\{0\},$$

as required. □

Using the above results we are now in a position to prove the main lemma of this section.

Proof of Theorem 2.14

Through the operator norm definition, we know

$$\|\mathcal{L}\|_{\mathscr{L}(L^2(V))} = \sup_{g\in L^2(V)} \left\{ \frac{\|\mathcal{L}g\|_{L^2(V)}}{\|g\|_{L^2(V)}} : g \neq 0 \right\}$$
$$= \sup_{g\in L^2(V)} \left\{ \left(\frac{\langle \mathcal{L}g, \mathcal{L}g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \right)^{1/2} : g \neq 0 \right\}$$
$$= \sup_{g\in L^2(V)} \left\{ \left(\frac{\langle g, \mathcal{L}^2 g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \right)^{1/2} : g \neq 0 \right\}. \tag{2.51}$$

Now using Lemmas 2.20 and 2.17 we know that for all $g \in L^2(V)\backslash\{0\}$

$$\frac{\langle g, \mathcal{L}^2 g\rangle_{L^2(V)}}{\langle g, g\rangle_{L^2(V)}} \leq 1,$$

and so the square root of this ratio is also bounded below one. Thus by (2.51) we have

$$\|\mathcal{L}\|_{\mathscr{L}(L^2(V))} \leq 1,$$

as required.

2.5.2 Convergence of Source Iteration

Using Theorem 2.14 we can now prove our main result concerning the convergence of source iteration with piecewise smooth cross sections. We first set up some notation that will be used. As stated in Section 2.2.1, the transport equation is given by

$$\mathcal{T}\psi(\mathbf{r},\Omega) = \sigma_S(\mathbf{r})\phi(\mathbf{r}) + Q(\mathbf{r}),$$

from which source iteration is given (in Algorithm 1) by

$$\mathcal{T}\psi^{(k+1)}(\mathbf{r},\Omega) = \sigma_S(\mathbf{r})\phi^{(k)}(\mathbf{r}) + Q(\mathbf{r}).$$

In both cases the boundary conditions are specified by (2.2). Subtracting these two equations gives

$$\mathcal{T}\left(\psi - \psi^{(k+1)}\right) = \sigma_S\left(\phi - \phi^{(k)}\right),$$

where now $\psi - \psi^{(k+1)}$ satisfies zero boundary conditions. Using Corollary 2.3 with $f = 0$ and $g = \sigma_S\left(\phi - \phi^{(k)}\right)$ we get

$$e^{(k+1)} = \mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r})e^{(k)} \tag{2.52}$$

where

$$e^{(k)} \equiv \phi - \phi^{(k)}. \tag{2.53}$$

Theorem 2.21:
Under the definitions above we have that

$$\left\|\sigma_T^{1/2}e^{(k+1)}\right\|_{L^2(V)} \leq \left\|\frac{\sigma_S}{\sigma_T}\right\|_{L^\infty(V)} \left\|\sigma_T^{1/2}e^{(k)}\right\|_{L^2(V)}, \tag{2.54}$$

and so since $\left\|\dfrac{\sigma_S}{\sigma_T}\right\|_{L^\infty(V)} < 1$ *the following limit holds*

$$\left\|e^{(k)}\right\|_{L^2(V)} \to 0, \quad as \quad k \to \infty.$$

Proof.
If we premultiply (2.52) by $\sigma_T^{1/2}$ we get

$$\begin{aligned}\sigma_T^{1/2} e^{(k+1)} &= \sigma_T^{1/2} \mathcal{K}_{\sigma_T} \sigma_S(\mathbf{r}) e^{(k)} \\ &= \sigma_T^{1/2} \mathcal{K}_{\sigma_T} \sigma_T^{1/2} \left(\frac{\sigma_S}{\sigma_T}\right) \sigma_T^{1/2} e^{(k)}\end{aligned}$$

Thus

$$\begin{aligned}\left\|\sigma_T^{1/2} e^{(k+1)}\right\|_{L^2(V)} &= \left\|\sigma_T^{1/2} \mathcal{K}_{\sigma_T} \sigma_T^{1/2} \left(\frac{\sigma_S}{\sigma_T}\right) \sigma_T^{1/2} e^{(k)}\right\|_{L^2(V)} \\ &\leq \left\|\sigma_T^{1/2} \mathcal{K}_{\sigma_T} \sigma_T^{1/2}\right\|_{\mathscr{L}(L^2(V))} \left\|\left(\frac{\sigma_S}{\sigma_T}\right) \sigma_T^{1/2} e^{(k)}\right\|_{L^2(V)} \\ &\leq \left\|\sigma_T^{1/2} \mathcal{K}_{\sigma_T} \sigma_T^{1/2}\right\|_{\mathscr{L}(L^2(V))} \left\|\frac{\sigma_S}{\sigma_T}\right\|_{L^\infty(V)} \left\|\sigma_T^{1/2} e^{(k)}\right\|_{L^2(V)}. \qquad (2.55)\end{aligned}$$

Using Theorem 2.14, equation (2.55) tells us that

$$\left\|\sigma_T^{1/2} e^{(k+1)}\right\|_{L^2(V)} \leq \left\|\frac{\sigma_S}{\sigma_T}\right\|_{L^\infty(V)} \left\|\sigma_T^{1/2} e^{(k)}\right\|_{L^2(V)}.$$

Now by (2.3) we know $\sigma_T(\mathbf{r}) > \sigma_S(\mathbf{r})$ for all $\mathbf{r} \in V$, and so

$$\left\|\frac{\sigma_S}{\sigma_T}\right\|_{L^\infty(V)} < 1.$$

Consequently

$$\left\|\sigma_T^{1/2} e^{(k)}\right\|_{L^2(V)} \to 0, \quad \text{as} \quad k \to \infty.$$

To finish the proof we note that σ_T is bounded over the domain V, and so we can immediately conclude

$$\left\|e^{(k)}\right\|_{L^2(V)} \to 0, \quad \text{as} \quad k \to \infty,$$

as required. □

This theorem proves that continuous source iteration, when applied to the neutron transport equation with piecewise smooth cross sections in space, converges to the scalar flux, ϕ. As in Section 2.5.1, this result holds in 1D, 2D and 3D.

An immediate consequence of this is the result proved rigorously in [67, Chapter 2]

that for constant cross sections (both in space and angle)

$$\left\| e^{(k)} \right\|_{L^2(V)} \leq \frac{\sigma_S}{\sigma_T} \left\| e^{(k-1)} \right\|_{L^2(V)},$$

however we will improve upon this bound in the next section.

2.5.3 Improved Bounds in the case of Constant Cross Sections

We showed in Theorem 2.21 that source iteration converges with piecewise smooth cross sections, and to do this we utilised a bound on the norm of the operator ${\sigma_T}^{1/2}\mathcal{K}_{\sigma_T}{\sigma_T}^{1/2}$. For constant cross sections this result was already known (see for example [67, Chapter 4]) and relies upon the operator norm bound $\|\mathcal{K}_{\sigma_T}\|_{\mathscr{L}(L^2(V))} \leq 1/\sigma_T$ (see [67, Theorem 2.9, p.29]). In this section we will prove a stricter bound on the norm of $\mathcal{K}_{\sigma_T}$ for constant cross sections, and thus obtain a tighter bound on the rate of convergence for source iteration in this instance.

Our first focus is on bounding the norm of $\mathcal{K}_{\sigma_T} : L^2(V) \to L^2(V)$ defined in (2.31) and (2.33) in 3D and 2D respectively. We will use the following definition.

Definition 2.22 (Diameter)**:**
The diameter *of a set $V \in \mathbb{R}^k$, $k = 2, 3$, is denoted* $\mathrm{diam}(V)$*, and is defined as follows*

$$\mathrm{diam}(V) \equiv \sup\{\|\mathbf{r} - \hat{\mathbf{r}}\| \ : \ \mathbf{r}, \hat{\mathbf{r}} \in V\}.$$

Using this concept, we are now able to state and prove the following result.

Theorem 2.23:
Let $V \subset \mathbb{R}^k$, $k = 2, 3$, be an open connected set and suppose all cross sections are strictly positive constants. Then the solution operator $\mathcal{K}_{\sigma_T} : L^2(V) \to L^2(V)$ defined in either (2.33) *or* (2.31) *satisfies*

$$\|\mathcal{K}_{\sigma_T}\|_{\mathscr{L}(L^2(V))} \leq \frac{1}{\sigma_T}\left[1 - \exp\left(-\sigma_T \mathrm{diam}(V)\right)\right] \tag{2.56}$$

where $\mathscr{L}(L^2(V))$ denotes the space of linear operators from $L^2(V)$ to $L^2(V)$.

Proof.
The argument in this proof will be presented for the case $V \subset \mathbb{R}^3$, however in two dimensional space the argument is similar. First let $\varphi \in L^2(V)$, then we know

$$\left(\mathcal{K}_{\sigma_T}\varphi\right)(\mathbf{r}) \equiv \int_V k_{\sigma_T}(\mathbf{r}, \mathbf{r}')\varphi(\mathbf{r}') \, \mathrm{d}\mathbf{r}',$$

where

$$k_{\sigma_T}(\mathbf{r},\mathbf{r}') \equiv \frac{\exp\left(-\sigma_T \left\|\mathbf{r}-\mathbf{r}'\right\|_2\right)}{4\pi \left\|\mathbf{r}-\mathbf{r}'\right\|_2^2}.$$

The kernel, k_{σ_T}, satisfies

$$\begin{aligned} k_{\sigma_T}(\mathbf{r},\mathbf{r}') &> 0, \\ k_{\sigma_T}(\mathbf{r},\mathbf{r}') &= k_{\sigma_T}(\mathbf{r}',\mathbf{r}), \end{aligned}$$

for all $\mathbf{r},\mathbf{r}' \in V$. To estimate the norm of $\mathcal{K}_{\sigma_T}$ we first write, for any $\varphi \in L^2(V)$

$$\begin{aligned} \left\|\mathcal{K}_{\sigma_T}\varphi\right\|_{L^2(V)}^2 &= \int_V \left| \int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\varphi(\mathbf{r}')\,\mathrm{d}\mathbf{r}' \right|^2 \,\mathrm{d}\mathbf{r} \\ &\leq \int_V \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')^{1/2} \left|\varphi(\mathbf{r}')\right| k_{\sigma_T}(\mathbf{r},\mathbf{r}')^{1/2}\,\mathrm{d}\mathbf{r}' \right)^2 \,\mathrm{d}\mathbf{r}. \end{aligned}$$

Then using the Cauchy-Schwarz inequality

$$\begin{aligned} \left\|\mathcal{K}_{\sigma_T}\varphi\right\|_{L^2(V)}^2 &\leq \int_V \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}') \left|\varphi(\mathbf{r}')\right|^2 \,\mathrm{d}\mathbf{r}' \right) \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \right) \,\mathrm{d}\mathbf{r} \\ &\leq \max_{\mathbf{r}\in V} \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \right) \int_V \int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}') \left|\varphi(\mathbf{r}')\right|^2 \,\mathrm{d}\mathbf{r}'\,\mathrm{d}\mathbf{r} \\ &= \max_{\mathbf{r}\in V} \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \right) \int_V \int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r} \left|\varphi(\mathbf{r}')\right|^2 \,\mathrm{d}\mathbf{r}' \\ &\leq \max_{\mathbf{r}\in V} \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \right) \max_{\mathbf{r}'\in V} \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r} \right) \int_V \left|\varphi(\mathbf{r}')\right|^2 \,\mathrm{d}\mathbf{r}' \\ &= \left[\max_{\mathbf{r}\in V} \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \right) \right]^2 \left\|\varphi\right\|_{L^2(V)}^2. \end{aligned} \tag{2.57}$$

Using (2.57) we therefore have that

$$\left\|\mathcal{K}_{\sigma_T}\right\|_{\mathscr{L}(L^2(V))} = \sup_{\substack{\varphi\in L^2(V) \\ \varphi\neq 0}} \frac{\left\|\mathcal{K}_{\sigma_T}\varphi\right\|_{L^2(V)}}{\left\|\varphi\right\|_{L^2(V)}} \leq \max_{\mathbf{r}\in V} \left(\int_V k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \right). \tag{2.58}$$

To get an estimate for (2.58) we will use spherical coordinates centred at $\mathbf{r}$, so

$$\mathbf{r}' = \mathbf{r} - s\Omega$$

where $\Omega \in \mathbb{S}^2$ and $s \in \mathbb{R}^+$. Since this gives $\|\mathbf{r} - \mathbf{r}'\|_2 = \|s\Omega\|_2 = s$, we have that

$$k_{\sigma_T}(\mathbf{r}, \mathbf{r}') = \frac{\exp(-\sigma_T s)}{4\pi s^2},$$

and so, defining $L \equiv \text{diam}(V)$, we obtain

$$\begin{aligned}\int_V k_{\sigma_T}(\mathbf{r}, \mathbf{r}')\, \mathrm{d}\mathbf{r}' &\leq \int_0^L \int_{\mathbb{S}^2} \frac{\exp(-\sigma_T s)}{4\pi s^2} s^2\, \mathrm{d}\Omega\, \mathrm{d}s \\ &= \int_0^L \exp(-\sigma_T s)\, \mathrm{d}s \\ &= \frac{1}{\sigma_T}\left[1 - \exp(-\sigma_T L)\right].\end{aligned}$$

Note that in the above an inequality is obtained by using the maximum diameter L, regardless of angle. Returning to (2.58) we can now conclude

$$\|\mathcal{K}_{\sigma_T}\|_{\mathscr{L}(L^2(V))} \leq \frac{1}{\sigma_T}\left[1 - \exp(-\sigma_T \text{diam}(V))\right],$$

as required. □

Next we prove an equivalent result in the 1D case where $\mathcal{K}_{\sigma_T}$ is defined in (2.35). To do this, we make note of the following useful integral.

Remark 2.24:

$$\int_0^1 \exp\left(\frac{-a}{x}\right) dx = E_2(a), \tag{2.59}$$

with $a > 0$, and where E_2 is the so-called exponential integral *function defined in Abramowitz and Stegun [1, p.228, (5.1.4)] to be*

$$E_2(z) \equiv \int_1^\infty \frac{e^{-zt}}{t^2}\, dt.$$

The integral (2.59) *is found by making the substitution $t = 1/x$.*

Note further this function satisfies $0 < E_2(x) < 1$ for all $x > 0$ ($E_2(x) = 1$ iff $x = 0$) and is a decreasing function for positive x (as can be verified using [1, (5.1.19),(5.1.23) and (5.1.26)]).

Theorem 2.25:

Let $V = (x_L, x_R) \subset \mathbb{R}$ be an open connected set and all cross sections are positive constants. Then the solution operator $\mathcal{K}_{\sigma_T} : L^2(V) \to L^2(V)$, as defined in (2.35),

satisfies

$$\|\mathcal{K}_{\sigma_T}\|_{\mathscr{L}(L^2(V))} \leq \frac{1}{\sigma_T}\left[1 - E_2\left(\sigma_T L\right)\right], \tag{2.60}$$

with diameter $L \equiv x_R - x_L$*, and where* $\mathscr{L}(L^2(V))$ *denotes the space of linear operators from* $L^2(V)$ *to* $L^2(V)$*.* E_2 *is an* exponential integral *function defined in Abramowitz and Stegun [1, p.228, (5.1.4)].*

Proof.
The first half of the proof follows the same logic as that of Theorem 2.23. First let $\varphi \in L^2(V)$, then we know

$$\left(\mathcal{K}_{\sigma_T}\varphi\right)(x) \equiv \int_V k_{\sigma_T}(x,y)\varphi(y)\,\mathrm{d}y,$$

where in 1D the kernel, k_{σ_T}, is defined to be

$$k_{\sigma_T}(x,y) \equiv \frac{1}{2}\int_0^1 \frac{1}{\mu}\exp\left(-\frac{\sigma_T}{\mu}\left|x-y\right|\right)\mathrm{d}\mu.$$

As in higher dimensions, the kernel satisfies

$$\begin{aligned} &k_{\sigma_T}(x,y) > 0, \\ &k_{\sigma_T}(x,y) = k_{\sigma_T}(y,x), \end{aligned}$$

for all $x, y \in V$. Taking the norm of $\mathcal{K}_{\sigma_T}\varphi$ we can use identical logic as was used to find (2.57), and obtain

$$\|\mathcal{K}_{\sigma_T}\varphi\|^2_{L^2(V)} \leq \left[\max_{x\in V}\left(\int_V k_{\sigma_T}(x,y)\,\mathrm{d}y\right)\right]^2 \|\varphi\|^2_{L^2(V)}. \tag{2.61}$$

Using (2.61) we therefore have that

$$\|\mathcal{K}_{\sigma_T}\|_{\mathscr{L}(L^2(V))} = \sup_{\substack{\varphi\in L^2(V)\\ \varphi\neq 0}} \frac{\|\mathcal{K}_{\sigma_T}\varphi\|_{L^2(V)}}{\|\varphi\|_{L^2(V)}} \leq \max_{x\in V}\left(\int_V k_{\sigma_T}(x,y)\,\mathrm{d}y\right), \tag{2.62}$$

which is the same as was obtained in the 3D case, (2.58).

To proceed from here we use the definition of the 1D kernel, (2.30), and find

$$\int_V k_{\sigma_T}(x,y)\,\mathrm{d}y = \frac{1}{2}\int_V \int_0^1 \frac{1}{\mu}\exp\left(-\frac{\sigma_T}{\mu}\left|x-y\right|\right)\mathrm{d}\mu\,\mathrm{d}y$$
$$= \frac{1}{2}\int_0^1 \int_{x_L}^{x_R} \frac{1}{\mu}\exp\left(-\frac{\sigma_T}{\mu}\left|x-y\right|\right)\mathrm{d}y\,\mathrm{d}\mu. \tag{2.63}$$

For $\mu \neq 0$ we can always write $y = x + s\mu$ for some $s \in \mathbb{R}$ (noting that this may be negative), and so

$$x - y = -s\mu.$$

Applying this as a change of variables, we continue from (2.63)

$$\int_V k_{\sigma_T}(x,y)\,\mathrm{d}y = \frac{1}{2}\int_0^1 \int_{(x_L-x)/\mu}^{(x_R-x)/\mu} \frac{1}{\mu}\exp\left(-\frac{\sigma_T}{\mu}\left|-s\mu\right|\right)\mu\,\mathrm{d}s\,\mathrm{d}\mu$$
$$= \frac{1}{2}\int_0^1 \int_{(x_L-x)/\mu}^{(x_R-x)/\mu} \exp\left(-\sigma_T\left|s\right|\right)\mathrm{d}s\,\mathrm{d}\mu. \tag{2.64}$$

Setting $l = \max\{|x_L - x|, |x_R - x|\}$ we continue from (2.64) and find

$$\int_V k_{\sigma_T}(x,y)\,\mathrm{d}y \leq \frac{1}{2}\int_0^1 \int_{-l/\mu}^{l/\mu} \exp\left(-\sigma_T\left|s\right|\right)\mathrm{d}s\,\mathrm{d}\mu$$
$$= \frac{1}{2}\int_0^1 2\int_0^{l/\mu} \exp\left(-\sigma_T s\right)\mathrm{d}s\,\mathrm{d}\mu$$
$$= \int_0^1 \frac{1}{\sigma_T}\left[1-\exp\left(-\sigma_T\frac{l}{\mu}\right)\right]\mathrm{d}\mu$$
$$= \frac{1}{\sigma_T}\left[1-\int_0^1 \exp\left(-\sigma_T\frac{l}{\mu}\right)\mathrm{d}\mu\right].$$

Now we can use the integral relation (2.59), which leaves

$$\int_V k_{\sigma_T}(x,y)\,\mathrm{d}y \leq \frac{1}{\sigma_T}\left[1-E_2\left(\sigma_T l\right)\right]. \tag{2.65}$$

To conclude this, we use the observation in Remark 2.24 that $E_2(x)$ is a decreasing function for positive x. Consequently (2.65) attains a maximum over $x \in [x_L, x_R]$ when x lies on either boundary, whereby $l = L \equiv x_R - x_L$. Therefore (2.62) becomes

$$\left\|\mathcal{K}_{\sigma_T}\right\|_{\mathscr{L}(L^2(V))} \leq \frac{1}{\sigma_T}\left[1-E_2\left(\sigma_T L\right)\right],$$

as required. □

The implications of this bound are not immediately clear, and so we take a moment here to understand it. As mentioned at the beginning of this section, the previous known bound on the operator norm of $\mathcal{K}_{\sigma_T}$ with constant cross sections was (in our notation) $\|\mathcal{K}_{\sigma_T}\|_{\mathscr{L}(L^2(V))} \leq 1/\sigma_T$ given by Scheben [67, Theorem 2.9, p.29]. Recalling Remark 2.24 we know that $E_2(x)$ is a decreasing, strictly positive function bounded below 1 for $x > 0$. Consequently $0 < 1 - E_2(\sigma_T L) < 1$ provided $\sigma_T L > 0$, which holds true for any physically relevant domain and material data. It is thus clear that our new bound for constant cross sections is always an improvement over the previous bound. Furthermore it has the greatest improvement for small values of $\sigma_T L$ since that corresponds to when the term in square brackets is smallest.

We can now write down a convergence result for source iteration that uses these new bounds. Following the same steps as Section 2.5.2 we can find the error equation for source iteration with constant cross sections

$$e^{(k+1)} = \sigma_S \mathcal{K}_{\sigma_T} e^{(k)} \tag{2.66}$$

where as before

$$e^{(k)} \equiv \phi - \phi^{(k)}.$$

With this we have the following convergence result

Corollary 2.26:
Under the definitions above, for $V \subset \mathbb{R}^n$, $n = 2, 3$,

$$\left\| e^{(k+1)} \right\|_{L^2(V)} \leq \frac{\sigma_S}{\sigma_T} \left[1 - \exp\left(-\sigma_T \mathrm{diam}(V) \right) \right] \left\| e^{(k)} \right\|_{L^2(V)}, \tag{2.67}$$

and for $V = [x_L, x_R] \subset \mathbb{R}$,

$$\left\| e^{(k+1)} \right\|_{L^2(V)} \leq \frac{\sigma_S}{\sigma_T} \left[1 - E_2\left(\sigma_T L \right) \right] \left\| e^{(k)} \right\|_{L^2(V)}, \tag{2.68}$$

where $L \equiv x_R - x_L$. Therefore in 1D, 2D and 3D it holds that

$$\left\| e^{(k)} \right\|_{L^2(V)} \to 0 \quad as \quad k \to \infty.$$

Proof.

In 1D, 2D and 3D we can take the norm of (2.66) and find

$$\begin{aligned}\left\|e^{(k+1)}\right\|_{L^2(V)} &= \sigma_S \left\|\mathcal{K}_{\sigma_T} e^{(k)}\right\|_{L^2(V)} \\ &\leq \sigma_S \left\|\mathcal{K}_{\sigma_T}\right\|_{\mathscr{L}(L^2(V))} \left\|e^{(k)}\right\|_{L^2(V)}\end{aligned}$$

Now using Theorems 2.23 and 2.25 we immediately obtain the required error norm bounds. From here the limit holds trivially. □

This result gives us a new, tighter bound on the rate of convergence of source iteration when assuming constant cross sections. Whilst this does not guarantee that a discrete implementation of source iteration will adhere to this rate, we might expect to observe such a rate for a sufficiently fine discretisation. Indeed in Section 2.7.1 we will observe even faster rates of convergence, indicating that this new result is still not telling us the whole story.

2.6 Benefits and Limitations of Source Iteration

Theorem 2.21 indicates that the rate of convergence of source iteration may be heavily governed by the maximum norm of the so-called *scattering ratio*, $\sigma_S(\mathbf{r})/\sigma_T(\mathbf{r})$. One of the main advantages of source iteration over other more advanced iterative methods is the low computational cost per iteration. For situations in which source iteration converges quickly this can make it a very computationally cheap method to use. One such situation is when the scattering ratio is small at all points of the spatial domain. Physically this corresponds to a situation in which the dominant outcome of neutron interactions is absorption in all regions in the spatial domain, i.e. all materials in the model have relatively high absorption cross sections.

Theorem 2.21 similarly highlights one of the main limitations of source iteration: a potential for slow convergence when the scattering ratio is close to one in any region of the spatial domain, no matter how small. This is quite inconvenient since most physical situations that are modelled will have at least some material in the domain that predominantly scatters neutrons during interactions.

Later we will discuss several ways in which this problem can be resolved. One of our main points of focus will be the diffusion synthetic acceleration method, which is a type of the more general synthetic acceleration methods. These methods involve adding an 'update' to the approximate scalar flux at each step of source iteration. The aim is to choose an update that is most accurate in situations when source iteration has the worst convergence, namely in scattering dominated regions. Such an update can

be found by utilising a specific diffusion equation that will be introduced in Chapter 3, leading to diffusion synthetic acceleration (see Section 3.3).

We will also look at a domain decomposition approach to solving the transport equation in Chapter 4. By breaking the domain into several regions it is possible to use different solution methods in each region. Consequently, it becomes possible to isolate the regions where source iteration performs poorly and use a more appropriate solution method only in that region, keeping source iteration elsewhere. The obvious advantage this has is in maintaining a lower computational cost. More effective methods require more calculations per iteration, and so limiting the size of the problem they have to tackle can help keep these costs low. Another less obvious advantage to the domain-decomposition approach is in dealing with issues caused by discontinuities, as will be seen in Chapter 4.

We will conclude this chapter by looking at some numerical results to highlight the features discussed above. As well as numerically confirming our theory, we will observe some interesting behaviour that is not predicted by Theorem 2.21, and will discuss what could be causing this behaviour.

2.7 Numerical Results

In this final section we will present the results of several 2D numerical experiments. We will consider the problem of numerically solving the neutron transport equation, subject to zero incoming boundary conditions, for a 2D spatial domain and a 1D angular domain (see Section 2.2.2). To do this we will discretise in space using discontinuous Galerkin finite elements (see [61], [42]) with uniform triangular elements, and M_x and M_y intervals in the x and y directions respectively (see Figure 5-1). We will also use discrete ordinates to discretise in angle (see [22], [32, Chapter 9]) choosing N angular points as detailed in Section 5.2.2. Using this process we produce a linear system of equations which can be solved numerically. A fully detailed account of this discretisation is given in Chapter 5 and some knowledge of that chapter will be assumed here, however we will not rely heavily upon it.

Our experiments will focus on numerically verifying discrete analogues of the two convergence results in this chapter: specifically Theorems 2.21 and 2.23. We will assume that our discretisation is good enough that the behaviour of these discrete versions implies similar behaviour of the continuous forms. This is by no means guaranteed, however by showing that different levels of spatial refinement result in consistent outcomes we can increase our confidence in this implication.

To verify these results we will consider a simple model problem over a square do-

main, however for each theorem we will independently vary different features of this problem. Firstly when looking at Theorem 2.23 we will use cross sections that are constant over the whole domain, and we will try independently varying the scattering ratio and the domain diameter. After that when looking at Theorem 2.21 we will use cross sections that are piecewise-constant over the spatial domain, and will vary the scattering ratio only in half of the domain. We will conclude in each case that our experiments support our theory but underestimate the potential rate of convergence.

2.7.1 Verifying Theorem 2.23

We focus first on numerically verifying Theorem 2.23, specifically we would like to see numerically that

$$\|\mathcal{K}_{\sigma_T}\|_{\mathscr{L}(L^2(V))} \leq \frac{1}{\sigma_T}\left[1 - \exp\left(-\sigma_T \text{diam}(V)\right)\right],$$

where the cross sections and source are constant in space and angle. We will try varying two quantities: the scattering ratio, σ_S/σ_T, and the domain diameter, $\text{diam}\,(V)$. In each case we will see that the bound given in Theorem 2.23 holds. To do this we notice that, from (2.67), the theoretical error ratio is given by

$$\frac{\left\|e^{(k+1)}\right\|_{L^2(V)}}{\left\|e^{(k)}\right\|_{L^2(V)}} \leq \frac{\sigma_S}{\sigma_T}\Big[1 - \exp\left(-\sigma_T \text{diam}(V)\right)\Big], \tag{2.69}$$

It is this bound on the ratio between the norms of successive errors that we will verify, calculated via the quadrature explained in Chapter 5.

First of all we fix the source and total cross section as $Q = 1$ and $\sigma_T = 1$, and fix the domain $V = [0,1] \times [0,1]$. Then by choosing the value of σ_A we vary the scattering ratio from 0.1 to 2.5. Recall that the the cross sections satisfy $\sigma_T = \sigma_S + \sigma_A$ for all $\mathbf{r} \in V$, so with $\sigma_T = 1$ the scattering ratio in each of our experiments is given by $\sigma_S = 1 - \sigma_A$. For each value of σ_A we run source iteration until it converges to a tolerance of 10^{-4}, or until we reach 25 iterations (whichever occurs sooner). We use an initial guess of zero, with spatial resolutions $M_x = M_y = 8$ and angular resolution $N = 23$. The choice of angular resolution is motivated by the error estimates in [42], who highlight that choosing an angular resolution that depends upon the spatial resolution is important. As a result of this we always choose $N = \left\lceil 8\sqrt{\max\left\{M_x, M_y\right\}}\right\rceil$, where the factor of 8 is included to ensure a reasonable angular resolution even for coarse spatial grids. More details on the choices made can be found in Chapter 5.

Previously we have assumed $0 < \sigma_S/\sigma_T < 1$ via strictly positive cross sections, however here we are taking σ_S/σ_T up to 2.5 by allowing $\sigma_A < 0$. This can be interpreted

Scattering ratio, σ_S/σ_T	Iterations to converge	Numerically observed error ratio	Ratio bound given by (2.69)
0.1	3	0.036	0.063
0.3	4	0.108	0.190
0.5	5	0.180	0.316
0.7	7	0.252	0.442
0.9	8	0.324	0.569
1.1	10	0.397	0.695
1.3	12	0.469	0.822
1.5	15	0.541	0.948
1.7	19	0.613	1.075
1.9	25	0.685	1.201
2.1	25	0.757	1.327
2.3	25	0.829	1.454
2.5	25	0.901	1.580

Table 2.1: Table of the number of iterations taken for source iteration to converge, the observed error ratio on convergence, and the new bound on this ratio, for varying values of the scattering ratio.

as a very basic inclusion of fission in the equation, and we mentioned this briefly when we initially simplified the transport equation (Section 1.1.1). If we instead restrict the fission cross section, σ_F, and the number of secondary neutrons produced, ν, to be constant (not zero) then the simplified transport equation becomes

$$\Omega \cdot \nabla\psi(\mathbf{r},\Omega) + \sigma_T\psi(\mathbf{r},\Omega) = (\sigma_S + \nu\sigma_F)\,\phi(\mathbf{r}) + Q.$$

Considering a scattering ratio greater than one is consequently a very basic inclusion of secondary neutrons from collisions. Since it can be interpreted as a physically relevant situation, and also yields interesting numerical results, we include it here.

In Table 2.1 for each value of the scattering ratio we have given: (column 2) the number of iterations taken to converge; (column 3) the observed error ratio at convergence; and (column 4) the theoretical bound on this ratio.

Immediately we notice that the observed ratio is always less that than the bound (given by (2.69)), supporting the result in Theorem 2.23. We also see that the bound is quite pessimistic, predicting that (for the chosen cross section and source values) source iteration may diverge for scattering ratios over c.1.6. Surprisingly in this example we see source iteration converging for ratios in excess of 2.5, suggesting our theory doesn't yet tell the whole story. In proving Theorem 2.23 we took several inequalities, which have clearly had an effect. In our example the error ratio is a constant scaling (c.0.57) of

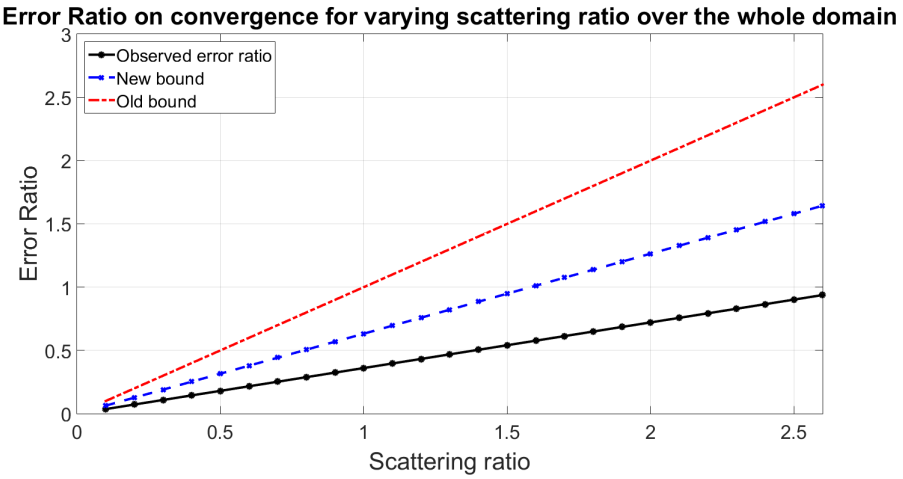

Figure 2-3: Plot of the observed error ratio (solid, black, dotted, bottom), the new bound on this ratio (blue, dashed, crossed, middle), and the old bound on this ratio (red, dashed, top).

the theoretical bound. Varying the spatial resolution does not influence this constant, however by changing either the domain diameter or the magnitude of the cross sections we are able to vary this scale between 0 and 1. Our theory already includes the effect of the domain diameter (and we will observe its effects in the next section) however there may be improvements to be made by involving the size of the total cross section and not just the scattering ratio. Lastly we remark that the ceiling of 25 iterations was artificially enforced and not an outcome of the numerics.

In Figure 2-3 we have plotted the last two columns of Table 2.1 against the corresponding scattering ratio (first column). We have also included the bound derived by F. Scheben, 2011, [67, Theorem 2.9] for comparison. This figure highlights the increasing pessimism of our bound as the scattering ratio increases, as well as the fixed ratio between columns 3 and 4 in Table 2.1. It also shows the improvement of our new bound over the previous known bound from the literature.

Next we will try varying the diameter of the spatial domain, V, whilst fixing the scattering ratio and source. We achieve this by maintaining a square domain, $(x, y) \in [0, D] \times [0, D]$, and letting D increase from 1 to 10. For each diameter we again run source iteration until it converges to a tolerance of 10^{-4} with an initial guess of zero, spatial resolutions $M_x = M_y = 8$ and angular resolution $N = 23$. We fix $\sigma_T = 1$, $\sigma_A = 0.5$ and $Q = 1$, and so the scattering ratio is always 0.5.

In Table 2.2 we have summarised the same data as in Table 2.1, however we have instead varied the domain diameter (column 1). Once again we observe that the theoretical bound holds in all tested cases, supporting the result of Theorem 2.23. We also

Domain diameter, diam (V)	Iterations to converge	Numerically observed error ratio	Ratio bound given by (2.69)
1.0	5	0.180	0.316
2.0	8	0.282	0.432
3.0	10	0.344	0.475
4.0	12	0.384	0.491
5.0	13	0.411	0.497
6.0	14	0.429	0.499
7.0	15	0.443	0.500
8.0	15	0.453	0.500
9.0	16	0.461	0.500
10.0	16	0.467	0.500

Table 2.2: Table of the number of iterations taken for source iteration to converge, the observed error ratio on convergence, and the new bound on this ratio, for varying domain diameters.

see that column 3 is no longer a constant scale of column 4. In fact for larger domain diameters our theoretical bound approaches the scattering ratio (0.5) and is closer to the observed error ratio. We can understand this by observing that increasing the diameter of the domain increases its *optical thickness* (the number of mean free paths that make up its width), and as a result the behaviour of the neutrons is closer to that of diffusion [54]. As we will discuss in the next chapter this reduces the effectiveness of source iteration and pushes its rate of convergence closer to the scattering ratio (as in the bound by F. Scheben, [67]). Trying to include the effects of optical thickness in Theorem 2.23 might yield a sharper bound.

2.7.2 Verifying Theorem 2.21

Our next aim is to numerically verify the result presented in Theorem 2.21. To do this we will consider the same problem as in Section 2.7.1 (over the square domain $V = [0,1] \times [0,1]$) however we will choose a piecewise constant absorption cross section, defined as

$$\sigma_A(\mathbf{r}) = \begin{cases} \sigma_{A1}, & \mathbf{r} \in V_1 \\ \sigma_{A2}, & \mathbf{r} \in V_2 \end{cases},$$

where $V_2 \equiv [0, 0.5] \times [0, 0.5] \cup [0.5, 1] \times [0.5, 1]$ and $V_1 \equiv V \backslash V_2$. Figure 2-4 contains a sketch of how the material properties change over the domain. We vary σ_{A2}, causing the scattering ratio in domain V_2 to vary between 0.1 and 4, whilst keeping the ratio

Scattering ratio in V_2, i.e. σ_{S2}/σ_T	Iterations to converge	Numerically observed error ratio	Ratio bound given by (2.70)
0.1	4	0.126	0.5
0.2	5	0.136	0.5
0.3	5	0.148	0.5
0.4	5	0.163	0.5
0.5	5	0.180	0.5
0.6	6	0.199	0.6
0.7	6	0.219	0.7
0.8	6	0.240	0.8
0.9	7	0.262	0.9
1.0	7	0.284	1.0
1.5	10	0.398	1.5
2.0	14	0.514	2.0
2.5	20	0.632	2.5
3.0	25	0.750	3.0
3.5	25	0.868	3.5
4.0	25	0.987	4.0

Table 2.3: Table of the number of iterations taken for source iteration to converge, the observed error ratio on convergence, and the new bound on this ratio, for varying values of the scattering ratio in half of the domain, with the ratio fixed as 0.5 elsewhere.

in V_1 fixed as 0.5 (by choosing $\sigma_{A1} = 0.5$). For each value we run source iteration until it converges to a tolerance of 10^{-4}, or until we reach 25 iterations (whichever occurs sooner). We will use the same spatial and angular resolution as in the previous section, and we fix $\sigma_T = Q = 1$. With a constant total cross section the multiples of ${\sigma_T}^{1/2}$ on either side of (2.54) cancel, so the bound in Theorem 2.21 becomes

$$\frac{\left\|e^{(k+1)}\right\|_{L^2(V)}}{\left\|e^{(k)}\right\|_{L^2(V)}} \leq \left\|\frac{\sigma_S}{\sigma_T}\right\|_{L^\infty(V)}. \tag{2.70}$$

To numerically verify this bound we solve the transport equation using source iteration and measure the observed error ratio between the two iterations immediately preceding convergence. The data from this experiment is in Table 2.3, where we have given (column 1) a range of values of the scattering ratio in V_2. For each value we have then listed: (column 2) the number of iterations taken to converge; (column 3) the observed error ratio at convergence; and (column 4) the theoretical bound on this ratio.

Straight away we observe that the theoretical bound is always greater than the observed ratio, supporting the conclusion of Theorem 2.21. As in Section 2.7.1 we see

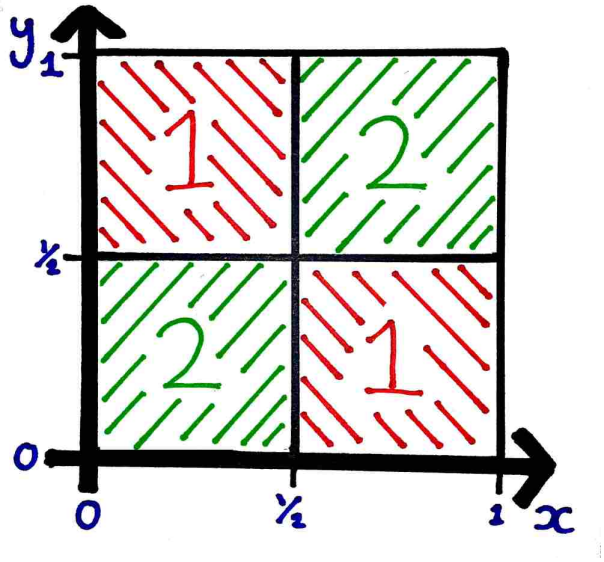

Figure 2-4: Figure illustrating the regions of the domain with different material properties.

that our bound is not strict, however we now find that the observed ratio is no longer a constant scale of the bound. In fact we see that as the scattering ratio in V_2 increases, the observed error ratio becomes a smaller proportion of the theoretical bound. This is not surprising since, while our bound only focusses on the 'worst case' scattering ratio, in reality half of the domain has a scattering ratio of 0.5 and so source iteration might be expected to perform well in that region. Being able to divide the spatial domain into these 'good' and 'bad' domains is a driving motivation behind our work in Chapter 4. More generally speaking, since our new bound, (2.70), is very similar to that of F. Scheben [67] (indeed it is equivalent for constant cross sections) it is not unexpected that we observe this lack of strictness. As in our earlier test we see that source iteration converges for scattering ratios in excess of one, though now this excess is only in half of the domain. In our earlier test source iteration began to diverge for scattering ratios over 2.5, however by restricting this excess to only half of the domain we have been able to observe convergence for ratios over 4.0. Interpreting this once again as a very basic inclusion of fission interactions in the model, we can understand that the higher levels of fission in V_2 are balanced out by the unreactive neighbouring domain, V_1.

Chapter 3

The Diffusion Approximation

Contents

3.1 Introduction 60

3.2 Asymptotic Derivation of the Diffusion Equation and Appropriate Boundary Conditions in 1D 62

3.2.1 Nondimensional Transport Equation 63

3.2.2 Outer Expansion: expanding away from the boundaries . . . 65

3.2.3 Inner Expansion: expanding near to the boundaries 70

3.2.4 Matching the Inner and Outer Expansions 74

3.2.5 Boundary Conditions for the Diffusion Approximation 75

3.2.6 Summary 77

3.3 Diffusion Synthetic Acceleration 79

3.3.1 Synthetic Acceleration 79

3.3.2 DSA Algorithm 83

3.3.3 DSA as a Preconditioner 86

3.4 Block Operator Diffusion 88

3.5 Numerical Results 101

3.5.1 Comparing SI and DSA 102

3.5.2 DSA in the Epsilon Limit 106

3.1 Introduction

In Chapter 2 we explored the convergence of an iterative method called source iteration. We showed, both theoretically (Section 2.5) and numerically (Section 2.7), that source iteration in all dimensions can converge slowly if the *scattering ratio* (a function

defined by the ratio between the scattering and total cross sections at each point in the domain) is close to one *anywhere* in the domain. For constant cross sections, we also demonstrated in Section 2.7.1 how increasing the domain diameter causes source iteration to converge more slowly, in agreement with the conclusion of Theorem 2.23.

In this chapter we will examine (primarily in 1D) a specific diffusion equation, the solution of which is an approximation to the scalar flux, ϕ. Crucially this approximation is most accurate under conditions that cause source iteration to converge poorly, namely when the domain is *optically thick* (see Section 3.2.1) and the neutron interactions are *scattering dominated*. It is this complementary feature that allows it to be used to accelerate the convergence of source iteration, leading to an iterative method which converges quickly in all situations. This method is known as *diffusion synthetic acceleration* or DSA, and a good discussion of the origins of this method can be found in Adams and Larsen [2, Chapter I.G].

The diffusion equation is used in most reactor core simulations [59]. Before being applied however, the domain is frequently homogenised by averaging the cross sections spatially ([69, Chapter 3, p.47]). This is because one of the conditions for the diffusion equation to be valid only holds away from material boundaries. Though this homogenisation is considered to be necessary, it is difficult to justify theoretically and the process itself is also not unique. An open question remains about finding an optimal homogenisation method [59, Section 8.2] and this is the source of one of the main limitations of the diffusion approximation, however accurate approximations can still be obtained by homogenising.

Unfortunately the conditions for the diffusion equation to be valid are also violated close to the domain boundaries, and this cannot be resolved by working on a homogenised problem. It is for this reason that a boundary layer analysis is required to obtain relevant boundary conditions for the diffusion equation [34], [41]. Consequently, as an approximation to the scalar flux, the diffusion equation is least accurate near the boundaries. This can be observed numerically, though we will not present such results here.

The literature contains different methods of deriving appropriate diffusion approximations. One method, which obtains the so-called P_1 approximation, works by expanding the neutron flux in terms of Legendre polynomials and truncating the expansion after the first two terms (see [71, Section 2.1.3], [57, p.123]). This also naturally results in the more complicated P_N equations (by truncating the expansion at higher order) which form a better approximation for higher N. The impracticality of utilising the P_N equations in higher dimensions lead to the development of the so-called *simplified* P_N (or SP_N) equations by E. M. Gelbard in 1960 [28]. A good discussion of the SP_N equa-

tions (including their history and two standard and asymptotic derivations) is given in [55].

In Section 3.2 we will use a matched asymptotic expansion method to derive an appropriate diffusion approximation along with suitable boundary conditions. The derivation of suitable boundary conditions is one of the main advantages of this method over the P_N method. This argument is not new and was first presented almost simultaneously in Habetler and Matkowsky, [34], and Larsen and Keller, [53]. Both of these papers start with the time dependent transport equation, whereas a follow up paper by Larsen, [49], worked from the steady-state transport equation, which was also considered later in [50] and [55]. Our work will include a non-fission neutron source term in the expansion, which was present in [53] but not in [34].

After this asymptotic work is completed, Section 3.3 uses the derived diffusion approximation to build the iterative method known as DSA as a two-step iterative method. From there we will move on in Section 3.3.3 to consider how DSA can instead be understood as a preconditioner to source iteration, using work by Faber and Manteuffel, [27], and Ashby et. al., [7]. Next in Section 3.4 we will introduce a new interpretation of the link between the transport equation and the diffusion equation, based on applying Gaussian elimination to a block operator form of the neutron transport equation. The results we present are not as general as the asymptotic derivation in Section 3.2 and they rely upon the asymptotic results from that section, however they provide a new way of understanding this well known relationship. As we will suggest in Chapter 6, future efforts to make this work more independent of the asymptotics could be very useful in developing our understanding of the diffusion equation and of DSA. To conclude the chapter we present and discuss several numerical tests in Section 3.5 comparing source iteration and diffusion synthetic acceleration.

3.2 Asymptotic Derivation of the Diffusion Equation and Appropriate Boundary Conditions in 1D

Different methods for obtaining the diffusion approximation to the neutron transport equation exist. The usual techniques in the literature use either spherical harmonics or an asymptotic expansion method: it is the latter which we focus on in this chapter. This method is explored in both Larsen and Keller's 1974 paper [53] and in Habetler and Matkowsky's 1975 paper [34] (and [40] explores the discrete case). Habetler and Matkowsky go a step further and use a so-called *boundary layer analysis* to derive suitable boundary conditions for the diffusion approximation. While these papers are comprehensive, they cover a lot of ground very quickly and as a consequence lose clarity

by requiring many gaps to be filled in before the logic can be repeated. Furthermore, Habetler and Matkowsky do not directly include a non-fission source term of neutrons but instead consider one retrospectively to their calculation. We feel therefore that a more thoroughly explained asymptotic derivation of the diffusion approximation and relevant boundary conditions would be a useful addition to the existing literature. Such a derivation is provided in this section, and we include a non-fission neutron source term explicitly.

Specifically, we will use an asymptotic expansion method to derive a diffusion equation whose argument is an approximation to the scalar flux, ϕ (defined in (2.6)). The derivation will also yield conditions for when this approximation is most accurate. Following this we will carry out a boundary layer analysis (see Sections 3.2.3 and 3.2.4) to obtain suitable boundary conditions for this approximation based on the conditions that are imposed on the neutron flux, ψ. This approach is necessary due to the existence of so-called *boundary layers* near either end of the domain in which the flux exhibits rapid variation, and allows us to reconcile the differences in behaviour inside and outside of these layers. The analysis utilizes asymptotic expansions of the transport equation both outside and inside the boundary layers (called the *outer* and *inner* expansions respectively), and then later requires that the two expansions *match* at their interface. This work is an expanded version of that in [34] with helpful explanation provided by Jonathan Evans [25] and with reference to E. J. Hinch [37].

3.2.1 Nondimensional Transport Equation

We begin by finding a nondimensional form of the simplified 1D transport equation and boundary conditions given in Section 2.2.3. We will carry out the asymptotic derivation of the diffusion approximation on this nondimensional version, and it will allow us to parametrise the behaviour of the system by a single dimensionless parameter, ϵ, defined later.

Firstly, we rewrite the 1D transport equation from Section 2.2.3 as follows

$$\frac{\mu}{\sigma_T(x)}\frac{\partial}{\partial x}\psi(x,\mu) + \psi(x,\mu) = \frac{c(x)}{2}\int_{[-1,1]} \psi(x,\mu)\ \mathrm{d}\mu + \frac{Q(x)}{\sigma_T(x)},$$

with $x \in [0, D]$, $\mu \in [-1, 1]$, and subject to boundary conditions

$$\begin{aligned} \psi(0,\mu) &= f_L(\mu), \quad \text{when } \mu > 0, \\ \psi(D,\mu) &= f_R(\mu), \quad \text{when } \mu < 0, \end{aligned}$$

where $c(x) \equiv \sigma_S(x)/\sigma_T(x)$. Without losing generality we have assumed $x_L = 0$ and $x_R = D$, in order to make our nondimensionalisation clearer. We denote by $\langle\psi\rangle$ a representative value of ψ, which may be obtained from the boundary conditions or other known data. Similarly $\langle\sigma_T\rangle$ denotes a representative value of the total cross section and may be an average over the domain, or other typical value, of the known function σ_T.

Using these we nondimensionalise our variables as follows

$$\begin{gathered} x = D\hat{x}, \quad \psi(x,\mu) = \langle\psi\rangle\hat{\psi}(\hat{x},\mu), \quad Q(x) = \langle\psi\rangle\sigma_T(x)\hat{Q}(\hat{x}), \\ \sigma_T(x) = \langle\sigma_T\rangle\hat{\sigma}_T(\hat{x}), \quad c(x) = \hat{c}(\hat{x}). \end{gathered}$$

With these our transport equation becomes

$$\frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\hat{\psi}(\hat{x},\mu) + \hat{\psi}(\hat{x},\mu) = \frac{\hat{c}(\hat{x})}{2}\int_{[-1,1]}\hat{\psi}(\hat{x},\mu)\,\mathrm{d}\mu + \hat{Q}(\hat{x}), \tag{3.1}$$

where now $\hat{x} \in [0,1]$, and we have introduced

$$\epsilon \equiv \frac{1}{\langle\sigma_T\rangle D}. \tag{3.2}$$

This important dimensionless parameter is the ratio between the typical *mean free path*, $1/\langle\sigma_T\rangle$, and the length scale of the domain, D. The mean free path is defined as the average distance a neutron will travel between successive collisions. Small values of ϵ imply the average distance between successive neutron collisions is very small relative to the width of the domain, in which case the domain is said to be *optically thick*. During the coming asymptotic work it will also be seen that small values of ϵ correspond to when the neutron interactions are *scattering dominated*, meaning that the scattering cross section is a high proportion of the total cross section. These two conditions make it clear why a domain where ϵ is small is referred to as *diffusive*.

We can also use the representative neutron flux value to conveniently scale the boundary condition functions as follows,

$$f_L(\mu) \equiv \langle\psi\rangle\hat{f}_L, \quad f_R(\mu) \equiv \langle\psi\rangle\hat{f}_R.$$

The boundary conditions for (3.1) are then given by

$$\begin{aligned} \hat{\psi}(0,\mu) &= \hat{f}_L(\mu), \quad \text{when } \mu > 0, \\ \hat{\psi}(1,\mu) &= \hat{f}_R(\mu), \quad \text{when } \mu < 0. \end{aligned} \tag{3.3}$$

At thermal energies, values of ϵ can lie in the range $10^{-5} \sim 10^{-3}$, and hence we use ϵ as a small asymptotic expansion variable when examining the behaviour of the

nondimensional flux, $\hat{\psi}$.

The ϵ multiple of the derivative in (3.1) means this is a singular perturbation problem, i.e. its behaviour differs greatly between $\epsilon > 0$ and $\epsilon = 0$ (see [37, Section 1.2]). For much of the spatial domain this first order derivative has only an $\mathcal{O}(\epsilon)$ effect on the transport equation. However in thin regions near the boundaries (known as *boundary layers*) this derivative is sufficiently large that it outweighs the limiting effect of the small multiple, ϵ. It is this difference in behaviour inside and outside of the boundary layers that motivates the matching of expansions in each region (see [37, Chapter 5]). When expanding inside the boundary layer we will use a *stretched* spatial parameter to allow the behaviour to be captured effectively.

3.2.2 Outer Expansion: expanding away from the boundaries

We will first look at the outer expansion where we consider (3.1) away from the spatial boundaries. To avoid notational clutter, we will drop the 'hat' notation though we are still working in the dimensionless case, namely (3.1), which we write here again as

$$\frac{\epsilon\mu}{\sigma_T(x)}\frac{\partial}{\partial x}\psi(x,\mu) + \psi(x,\mu) = \frac{c(x)}{2}\int_{[-1,1]} \psi(x,\mu)\,\mathrm{d}\mu + Q(x), \tag{3.4}$$

subject to

$$\begin{aligned} \psi(0,\mu) &= f_L(\mu), \quad \text{when } \mu > 0, \\ \psi(1,\mu) &= f_R(\mu), \quad \text{when } \mu < 0. \end{aligned} \tag{3.5}$$

We will use asymptotic expansions of the neutron flux, ψ, and the functions c and Q in terms of ϵ. By doing this we will obtain a diffusion equation governing the behaviour of the scalar flux, ϕ, up to $\mathcal{O}(\epsilon^2)$ away from the boundary layers (see (3.31)). We begin by expanding as follows

$$\begin{aligned} \psi &\sim \psi_0 + \epsilon\psi_1 + \epsilon^2\psi_2 + \dots \\ c &\sim c_0 + \epsilon c_1 + \epsilon^2 c_2 + \dots \\ Q &\sim Q_0 + \epsilon Q_1 + \epsilon^2 Q_2 + \dots \end{aligned} \tag{3.6}$$

where $\psi_i \equiv \psi_i(x,\mu)$, $c_i \equiv c_i(x)$, and $Q_i \equiv Q_i(x)$. We follow the process of substituting these expansions into the nondimensional transport equation, (3.4), and then equating successive powers of ϵ. Our aim is to resolve the coefficients to leading order, and to obtain an approximate equation for the angularly independent part of the flux, ψ.

Firstly, consider the coefficient of the zero-th power of ϵ,

$$\epsilon^0: \quad \psi_0(x,\mu) = \frac{c_0(x)}{2}\int_{[-1,1]} \psi_0(x,\mu)\,\mathrm{d}\mu + Q_0(x). \tag{3.7}$$

We can immediately deduce that ψ_0 is independent of μ, i.e.

$$\psi_0(x,\mu) \equiv \psi_0(x), \tag{3.8}$$

thus (3.7) becomes

$$(1 - c_0(x))\psi_0(x) = Q_0(x). \tag{3.9}$$

If we allow $c_0(x) \not\equiv 1$ then the leading flux term, ψ_0, would just be a multiple of the leading source term, Q_0. Therefore in the epsilon limit, the dominant behaviour of the neutron flux would be governed by the source. To avoid this we require that $c_0(x) \equiv 1$, and so $Q_0(x) \equiv 0$.

Next we consider the coefficient of the first power of ϵ,

$$\epsilon^1: \quad \frac{\mu}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(x) + \psi_1(x,\mu) = \frac{1}{2}\int_{[-1,1]} \psi_1(x,\mu)\,\mathrm{d}\mu + c_1(x)\psi_0(x) + Q_1(x). \tag{3.10}$$

Equation (3.10) implies that ψ_1 is linear in μ and so we write

$$\psi_1(x,\mu) = \psi_{10}(x) + \mu\psi_{11}(x). \tag{3.11}$$

Substituting this into (3.10), we can equate the coefficients of different powers of μ, and after cancellation find

$$\mu^0: \qquad 0 = c_1(x)\psi_0(x) + Q_1(x) \tag{3.12}$$

$$\mu^1: \qquad 0 = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(x) + \psi_{11}(x). \tag{3.13}$$

Similarly to above, we wish to avoid forcing the leading order flux term, ψ_0, to be a scale of the source term in the ϵ limit. Consequently we set $c_1(x) \equiv 0$, and so $Q_1(x) \equiv 0$. With no new information from (3.12), the leading term ψ_0 is still unknown.

Continuing, the coefficient of the second power of ϵ gives us

$$\epsilon^2: \quad \frac{\mu}{\sigma_T(x)}\frac{\partial}{\partial x}\psi_1(x,\mu) + \psi_2(x,\mu) = \frac{1}{2}\int_{[-1,1]} \psi_2(x,\mu)\,\mathrm{d}\mu + c_2(x)\psi_0(x) + Q_2(x). \tag{3.14}$$

With ψ_1 linear in μ, we see that ψ_2 is quadratic in μ. Thus we write

$$\psi_2(x,\mu) = \psi_{20}(x) + \mu\psi_{21}(x) + \mu^2\psi_{22}(x),$$

and equating powers of μ in (3.14) then yields

$$\mu^0 : \qquad 0 = \frac{1}{3}\psi_{22}(x) + c_2(x)\psi_0(x) + Q_2(x) \tag{3.15}$$

$$\mu^1 : \qquad 0 = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_{10}(x) + \psi_{21}(x) \tag{3.16}$$

$$\mu^2 : \qquad 0 = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_{11}(x) + \psi_{22}(x). \tag{3.17}$$

Combining (3.17) and (3.13) gives

$$\psi_{22}(x) = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(x)\right), \tag{3.18}$$

which, when combined with (3.15), yields

$$0 = \frac{1}{3\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(x)\right) + c_2(x)\psi_0(x) + Q_2(x), \tag{3.19}$$

which is a diffusion equation for ψ_0. This implies that the nondimensional source term must be $\mathcal{O}(\epsilon^2)$ for it not to dominate the flux.

We finally look at the coefficient of the third power of ϵ,

$$\epsilon^3 : \quad \frac{\mu}{\sigma_T(x)}\frac{\partial}{\partial x}\psi_2(x,\mu) + \psi_3(x,\mu) = \frac{1}{2}\int_{[-1,1]} \psi_3(x,\mu)\,\mathrm{d}\mu + c_2(x)\psi_{10}(x) + c_3(x)\psi_0(x) + Q_3(x). \tag{3.20}$$

In the same way as before we see that ψ_3 is cubic in μ, and so we write

$$\psi_3(x,\mu) = \psi_{30}(x) + \mu\psi_{31}(x) + \mu^2\psi_{32}(x) + \mu^3\psi_{33}(x).$$

Equating coefficients of the powers of μ we get

$$\mu^0 : \qquad 0 = \frac{1}{3}\psi_{32}(x) + c_2(x)\psi_{10}(x) + c_3(x)\psi_0(x) + Q_3(x) \tag{3.21}$$

$$\mu^1 : \qquad 0 = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_{20}(x) + \psi_{31}(x) \tag{3.22}$$

$$\mu^2 : \qquad 0 = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_{21}(x) + \psi_{32}(x) \tag{3.23}$$

$$\mu^3 : \qquad 0 = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_{22}(x) + \psi_{33}(x). \tag{3.24}$$

Combining (3.23) with (3.16) we find

$$\psi_{32}(x) = \frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_{10}(x)\right). \tag{3.25}$$

Using this along with (3.21) we are left with

$$0 = \frac{1}{3\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_{10}(x)\right) + c_2(x)\psi_{10}(x) + c_3(x)\psi_0(x) + Q_3(x), \tag{3.26}$$

which is a diffusion equation for ψ_{10}.

We can bring together (3.19) and (3.26) to obtain

$$0 = \frac{1}{3\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\phi_{[1]}(x)\right) + c_2(x)\phi_{[1]}(x) + \epsilon c_3(x)\psi_0(x) + \big[Q_2 + \epsilon Q_3\big](x), \tag{3.27}$$

where

$$\phi_{[1]} \equiv \psi_0 + \epsilon\psi_{10}. \tag{3.28}$$

If we were to continue this asymptotic expansion and equating of coefficients we would obtain higher order terms. This would consolidate the terms with $c_i(x)$ coefficients, however it would also introduce higher order derivatives. Instead we stop at this second-order diffusion equation, (3.27), albeit with as yet unspecified coefficients and boundary conditions.

We return for the moment to the dimensional notation of Section 3.2.1, and recall that we defined the coefficient $c(x) = \sigma_S(x)/\sigma_T(x) = \hat{c}(\hat{x})$. Subsequently we have resolved an asymptotic expansion of the dimensionless $\hat{c}$ up to order ϵ^2, leaving

$$\hat{c}(\hat{x}) \sim 1 + \epsilon^2\hat{c}_2(\hat{x}) + \mathcal{O}(\epsilon^3).$$

Using the relation $\sigma_S(x) = \sigma_T(x) - \sigma_A(x)$ along with $\sigma_T(x) = \langle\sigma_T\rangle\hat{\sigma}_T(\hat{x})$ and $\epsilon =$

$1/(\langle\sigma_T\rangle D)$, it follows that

$$\hat{c}(\hat{x}) = \frac{\sigma_S(x)}{\sigma_T(x)} = 1 - \frac{\epsilon D\sigma_A(x)}{\hat{\sigma}_T(\hat{x})} \sim 1 + \epsilon^2 c_2(x) + \mathcal{O}(\epsilon^3).$$

This implies $\sigma_A(x) \sim \mathcal{O}(\epsilon)$, and a simple choice (also used and justified physically in [34] and elsewhere) is

$$D\sigma_A(x) = \epsilon\hat{\sigma}_A(\hat{x}), \tag{3.29}$$

which when combined with our expansion leaves

$$\hat{c}(\hat{x}) = 1 - \epsilon^2\frac{\hat{\sigma}_A(\hat{x})}{\hat{\sigma}_T(\hat{x})}. \tag{3.30}$$

Now returning to (3.27), using (3.30) (and again dropping the hat notation) we have

$$0 = \frac{1}{3}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\phi_{[1]}(x)\right) - \sigma_A(x)\phi_{[1]}(x) + \sigma_T(x)\frac{Q(x)}{\epsilon^2}, \tag{3.31}$$

which is a nondimensional diffusion equation for the unknown $\phi_{[1]}(x) = \psi_0(x)+\epsilon\psi_{10}(x)$.

To understand exactly how the argument of this diffusion equation relates to the neutron flux we write the flux expansion in full as follows

$$\begin{aligned}\psi(x,\mu) = \quad & \big[\psi_0(x) + \epsilon\psi_{10}(x) + \epsilon^2\psi_{20}(x) + \dots\big] \\ + \quad \mu\big[\quad & \epsilon\psi_{11}(x) + \epsilon^2\psi_{21}(x) + \dots\big] \\ + \quad \mu^2\big[\quad & \epsilon^2\psi_{22}(x) + \dots\big] \\ + \dots.\end{aligned}$$

Averaging over angle to obtain the scalar flux gives

$$\phi(x) = \psi_0(x) + \epsilon\psi_{10}(x) + \epsilon^2\left[\dots\right] + \dots, \tag{3.32}$$

and we see that $\phi_{[1]}$, the solution of (3.31), is accurate to $\mathcal{O}(\epsilon^2)$.

Note that the small ϵ multiple on the derivative in (3.4) means that, for much of the domain (namely the interior) the behaviour arising from that term does not influence the behaviour of the flux. However near the boundaries this leads to so-called *boundary layers* in which the behaviour needs to be recovered, and to do this we need to carry out an asymptotic expansion near the boundary. This is commonly called an *inner expansion* and will be our focus for the next section.

3.2.3 Inner Expansion: expanding near to the boundaries

We focus now on asymptotically expanding the non-dimensional transport equation, (3.4), inside of a boundary layer near $x = 0$. The equivalent analysis in the boundary layer near $x = 1$ follows a very similar argument.

In the boundary layer near $x = 0$, the flux, ψ, increases rapidly enough in space to overcome the small ϵ multiple on the spatial derivative term. In order to be able to satisfy any boundary conditions we must capture this behaviour. To do this we scale the spatial variable as

$$x = \epsilon y,$$

so we are working in the domain $y \in [0, 1/\epsilon]$. (For the analysis near $x = 1$, an appropriate scaling is $x = 1 - \epsilon y$). In terms of this 'stretched' variable, we define

$$\psi(\epsilon y, \mu) \equiv \Psi(y, \mu),$$

and the transport equation (3.4) becomes

$$\frac{\mu}{\sigma_T(\epsilon y)} \frac{\partial}{\partial y} \Psi(y, \mu) + \Psi(y, \mu) = \frac{c(\epsilon y)}{2} \int_{[-1,1]} \Psi(y, \mu) \, \mathrm{d}\mu + Q(\epsilon y). \tag{3.33}$$

We will carry over our results from Section 3.2.2 concerning c and Q. We will assume that in the thin boundary layer the cross sections and source are essentially constant, and indeed they are at least slowly varying in our new scale [34]. Treating them as such, we set

$$\sigma_T(\epsilon y) \equiv \sigma_T(0), \quad Q(\epsilon y) \equiv \epsilon^2 Q_2(0) + \mathcal{O}(\epsilon^3).$$

Also, Taylor expanding the functions c_i about 0, $c(\epsilon y)$ becomes

$$c(\epsilon y) = 1 + \epsilon^2 c_2(0) + \epsilon^3 \left[c_3(0) + y c_2'(0) \right] + \mathcal{O}(\epsilon^4).$$

Using these and ignoring terms of order ϵ^3 or higher, (3.33) becomes

$$\frac{\mu}{\sigma_T(0)} \frac{\partial}{\partial y} \Psi(y, \mu) + \Psi(y, \mu) = \frac{1 + \epsilon^2 c_2(0)}{2} \int_{[-1,1]} \Psi(y, \mu) \, \mathrm{d}\mu + \epsilon^2 Q_2(0). \tag{3.34}$$

We are now working over the stretched domain $y \in [0, 1/\epsilon]$ and with $\mu \in [-1, 1]$. The boundary condition (3.5) at $y = 0$ becomes

$$\Psi(0,\mu) = f_L(\mu), \qquad \text{for } \mu > 0. \tag{3.35}$$

As in the outer case, we pose an expansion of the flux as

$$\Psi \sim \Psi_0 + \epsilon\Psi_1 + \epsilon^2\Psi_2 + \dots \tag{3.36}$$

where $\Psi_i \equiv \Psi_i(y,\mu)$. Substituting this into (3.34) and equating coefficients of increasing powers of epsilon yields

$$\epsilon^0 : \qquad \frac{\mu}{\sigma_T(0)}\frac{\partial}{\partial y}\Psi_0(y,\mu) + \Psi_0(y,\mu) = \frac{1}{2}\int_{[-1,1]} \Psi_0(y,\mu)\ \mathrm{d}\mu, \tag{3.37}$$

$$\epsilon^1 : \qquad \frac{\mu}{\sigma_T(0)}\frac{\partial}{\partial y}\Psi_1(y,\mu) + \Psi_1(y,\mu) = \frac{1}{2}\int_{[-1,1]} \Psi_1(y,\mu)\ \mathrm{d}\mu, \tag{3.38}$$

$$\begin{aligned}\epsilon^2 : \qquad \frac{\mu}{\sigma_T(0)}\frac{\partial}{\partial y}\Psi_2(y,\mu) + \Psi_2(y,\mu) &= \frac{1}{2}\int_{[-1,1]} \Psi_2(y,\mu)\ \mathrm{d}\mu \\ &\quad + \frac{c_2(0)}{2}\int_{[-1,1]} \Psi_0(y,\mu)\ \mathrm{d}\mu + Q_2(0).\end{aligned} \tag{3.39}$$

Similarly substituting the flux expansion into the boundary condition (3.35) we find, for $\mu > 0$

$$\Psi_0(0,\mu) = f_L(\mu), \tag{3.40}$$

$$\Psi_1(0,\mu) = 0, \tag{3.41}$$

$$\Psi_2(0,\mu) = 0. \tag{3.42}$$

We note here that the source term only begins to have an influence at $\mathcal{O}(\epsilon^2)$, and so does not affect the Ψ_0 or Ψ_1 equations. At this point, we require the following lemma.

Lemma 3.1:
Homogeneous integro-differential equations (3.37) *and* (3.38)*, which are of the form*

$$\frac{\mu}{\sigma_T(0)}\frac{\partial}{\partial y}\bar{\Psi}(y,\mu) + \bar{\Psi}(y,\mu) = \frac{1}{2}\int_{[-1,1]} \bar{\Psi}(y,\mu)\ d\mu,$$

have general solution

$$\bar{\Psi}(y,\mu) = a + b(\sigma_T(0)y - \mu) + \int_{-1}^{1} A(\nu)M_\nu(\mu)e^{\frac{-\sigma_T(0)y}{\nu}}\ d\nu, \tag{3.43}$$

where a and b are constants, and A a function, to be determined. In this,

$$M_\nu(\mu) = \frac{1}{2} P_r \frac{\nu}{\nu - \mu} + \lambda(\nu)\delta(\nu - \mu), \tag{3.44}$$

with

$$\lambda(\nu) = 1 - \nu \tanh^{-1}(\nu), \tag{3.45}$$

in which P_r denotes the Cauchy principle value and δ is the Dirac-delta function.

Proof.
This result is derived in Bell and Glasstone, [12, Section 2.2], but we also repeat it in Appendix A.1 for convenience. □

Using this lemma we know that the solution to (3.37) is given by

$$\Psi_0(y, \mu) = a_0 + b_0(\sigma_T(0)y - \mu) + \int_{[-1,1]} A_0(\nu) M_\nu(\mu) e^{-\frac{\sigma_T(0)y}{\nu}} \, \mathrm{d}\nu. \tag{3.46}$$

If we now apply the boundary condition at $y = 0$, given by (3.40), to (3.46) we obtain

$$f_L(\mu) = a_0 - \mu b_0 + \int_{[-1,1]} A_0(\nu) M_\nu(\mu) \, \mathrm{d}\nu, \quad \mu > 0. \tag{3.47}$$

To determine a_0 and b_0 we need to introduce two new functions and an associated orthogonality relation. First we define

$$\gamma(\mu) \equiv \frac{3\mu}{2X(-\mu)}, \tag{3.48}$$

where

$$X(z) \equiv \frac{1}{1 - z} \exp\left[\frac{1}{\pi} \int_{[0,1]} \frac{1}{\mu' - z} \tan^{-1}\left(\frac{\pi\mu'}{2\left(1 - \mu' \tanh^{-1}(\mu')\right)}\right) \mathrm{d}\mu'\right], \tag{3.49}$$

and also define

$$\gamma^j \equiv \int_{[0,1]} \mu^j \gamma(\mu) \, \mathrm{d}\mu. \tag{3.50}$$

These three definitions can be found in [20] as equations (2), (33) and (5b) on pages 131, 130 and 164 respectively. The function $X(z)$ is chosen to satisfy a number of conditions given in [20, Section 4.8], and in particular is such that γ satisfies the following relation,

$$\int_{[0,1]} M_\nu(\mu)\gamma(\mu)\,\mathrm{d}\mu = 0. \tag{3.51}$$

This equation can be found as equation (2) in [20, Section 6.9]. To make use of this we observe that for $\nu < 0$, (3.46) contains a growing exponential term which is not shared by the diffusion equation, (3.19). Consequently for these solutions to match in some overlapping region we require

$$A_0(\nu) = 0, \quad \nu < 0. \tag{3.52}$$

Multiplying (3.47) by $\gamma(\mu)$ we obtain, for $\mu > 0$

$$f_L(\mu)\gamma(\mu) = a_0\gamma(\mu) - \mu b_0\gamma(\mu) + \int_{[0,1]} A_0(\nu)M_\nu(\mu)\gamma(\mu)\,\mathrm{d}\nu,$$

whereby integrating over positive μ and utilising (3.51) we find

$$\int_{[0,1]} f_L(\mu)\gamma(\mu)\,\mathrm{d}\mu = a_0\gamma^0 - b_0\gamma^1. \tag{3.53}$$

Therefore

$$a_0 = \frac{b_0\gamma^1}{\gamma^0} + \frac{1}{\gamma^0}\int_{[0,1]} f_L(\mu)\gamma(\mu)\,\mathrm{d}\mu. \tag{3.54}$$

which is an expression for the unknown a_0 in terms of b_0. An expression for $A_0(\nu)$ can also be found by multiplying (3.47) by $\gamma(\mu)M_\nu(\mu)$ and integrating (see [34, equation (4.27)] and preceding derivation).

In a similar manner, the solution to (3.38) is given by

$$\Psi_1(y,\mu) = a_1 + b_1(\sigma_T(0)y - \mu) + \int_{[-1,1]} A_1(\nu)M_\nu(\mu)e^{-\frac{\sigma_T(0)y}{\nu}}\,\mathrm{d}\nu. \tag{3.55}$$

Using the same procedure as for (3.54) with the correct (zero) boundary condition given in (3.41) we find that

$$a_1 = \frac{b_1\gamma^1}{\gamma^0}. \tag{3.56}$$

In summary, (3.54) gives a_0 in terms of b_0, and so by (3.46) we also have Ψ_0 in terms of b_0. Similarly, (3.56) gives a_1 in terms of b_1, and so (3.55) gives us Ψ_1 in terms of b_1. In the next section we use our knowledge about the solution in the outer layer to find the constants b_0 and b_1 via a process known as *matching*.

3.2.4 Matching the Inner and Outer Expansions

At this point we have two asymptotic expansions for the neutron flux: an outer expansion for fixed x, (3.6), and an inner expansion for fixed y, (3.36). In this section we will carry out a procedure called *matching*, in which we will assume that the two expansions are equal in some overlapping region, where $\epsilon \ll x = \epsilon y \ll 1$. We will follow Van Dyke's matching rule (see [37, Section 1.5]) and aim to determine b_0 and b_1 such that this overlap equality occurs.

Employing a Taylor expansion about $x = 0$, the outer expansion (3.6) becomes

$$\begin{aligned}\psi(x,\mu) = & \left[\psi_0(0) + x\frac{\partial}{\partial x}\psi_0(0) + \frac{x^2}{2}\frac{\partial^2}{\partial x^2}\psi_0(0) + \dots\right] \\ & + \epsilon\left[\psi_1(0,\mu) + x\frac{\partial}{\partial x}\psi_1(0,\mu) + \frac{x^2}{2}\frac{\partial^2}{\partial x^2}\psi_1(0,\mu) + \dots\right] \\ & + \epsilon^2\left[\psi_2(0,\mu) + x\frac{\partial}{\partial x}\psi_2(0,\mu) + \frac{x^2}{2}\frac{\partial^2}{\partial x^2}\psi_2(0,\mu) + \dots\right] \\ & + \dots\end{aligned}$$

Fixing $x = \epsilon y$ and gathering like powers of epsilon we are left with

$$\begin{aligned}\psi(\epsilon y,\mu) = & \left[\psi_0(0)\right] \\ & + \epsilon\left[\psi_1(0,\mu) + y\frac{\partial}{\partial x}\psi_0(0)\right] \\ & + \epsilon^2\left[\psi_2(0,\mu) + y\frac{\partial}{\partial x}\psi_1(0,\mu) + \frac{y^2}{2}\frac{\partial^2}{\partial x^2}\psi_0(0)\right] \\ & + \dots\end{aligned} \tag{3.57}$$

The inner expansion, (3.36), is given as

$$\psi(\epsilon y,\mu) \equiv \Psi(y,\mu) = \Psi_0(y,\mu) + \epsilon\Psi_1(y,\mu) + \epsilon^2\Psi_2(y,\mu) + \dots \tag{3.58}$$

For fixed $x(\neq 0)$ in the limit as $\epsilon \to 0$ we know that $y \to \infty$, and so Ψ_0 and Ψ_1 (from (3.46) and (3.55) respectively) behave as

$$\Psi_0(y,\mu) \sim a_0 + b_0\left(\sigma_T(0)y - \mu\right), \tag{3.59}$$

$$\Psi_1(y,\mu) \sim a_1 + b_1\left(\sigma_T(0)y - \mu\right). \tag{3.60}$$

If we match the zero-th power of epsilon in (3.57) and (3.58) and use (3.59) we find

that

$$\psi_0(0) = \Psi_0(y,\mu) = a_0 + b_0\left(\sigma_T(0)y - \mu\right),$$

thus

$$a_0 = \psi_0(0), \quad b_0 = 0. \tag{3.61}$$

Next, matching the first power of epsilon in (3.57) and (3.58) and using (3.60) we find that

$$\psi_1(0,\mu) + y\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(0) = \Psi_1(y,\mu) = a_1 + b_1\left(\sigma_T(0)y - \mu\right).$$

Comparing dependencies on y we obtain

$$a_1 - b_1\mu = \psi_1(0,\mu), \tag{3.62}$$

$$b_1 = \frac{1}{\sigma_T(0)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(0). \tag{3.63}$$

Substituting (3.11), i.e. the expansion $\psi_1 \equiv \psi_{10} + \mu\psi_{11}$, into (3.62) and comparing μ dependencies, we are left with

$$a_1 = \psi_{10}(0). \tag{3.64}$$

Note here that knowing (3.56) and (3.63) we also have that

$$a_1 = \frac{\gamma^1}{\sigma_T(0)\gamma^0}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(0). \tag{3.65}$$

After this matching process, we now have expressions for the constants a_i and b_i, for $i = 0, 1$, using terms from the outer expansion evaluated at the boundary $x = 0$.

3.2.5 Boundary Conditions for the Diffusion Approximation

In this section we will use the inner and outer solutions (and associated constants) derived in the previous sections to obtain suitable boundary conditions for the diffusion equation (3.31).

Going back to the outer asymptotic expansion (3.32) for the neutron flux, ψ, once more we have

$$\begin{aligned}\psi(x,\mu) &= \psi_0(x) + \epsilon\psi_1(x,\mu) + \mathcal{O}(\epsilon^2) \\ &= \psi_0(x) + \epsilon\left[\psi_{10}(x) + \mu\psi_{11}(x)\right] + \mathcal{O}(\epsilon^2). \end{aligned} \tag{3.66}$$

Using our expressions for $\phi_{[1]}(x)$ and $\psi_{11}(x)$ (i.e. (3.28) and (3.13)) this tells us that

$$\psi(x,\mu) = \phi_{[1]}(x) - \frac{\epsilon\mu}{\sigma_T(0)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(x) + \mathcal{O}(\epsilon^2), \tag{3.67}$$

which is essentially equation (2.2) from [40]. Working from (3.66) again but setting $x = 0$ and using (3.61) and (3.62) we find

$$\psi(0,\mu) = a_0 + \epsilon\left[a_1 - b_1\mu\right] + \mathcal{O}(\epsilon^2),$$

whereby using the definitions of a_0, b_0, a_1 and b_1 ((3.54), (3.61), (3.65) and (3.63) respectively) this results in

$$\psi(0,\mu) = \frac{1}{\gamma^0}\int_{[0,1]} f_L(\mu)\gamma(\mu)\,\mathrm{d}\mu + \epsilon\left[\frac{\gamma^1}{\sigma_T(0)\gamma^0}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(0) - \frac{\mu}{\sigma_T(0)}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(0)\right] + \mathcal{O}(\epsilon^2). \tag{3.68}$$

If we equate (3.68) and (3.67) with $x = 0$, we find

$$\phi_{[1]}(0) = \frac{1}{\gamma^0}\int_{[0,1]} f_L(\mu)\gamma(\mu)\,\mathrm{d}\mu + \frac{\epsilon\gamma^1}{\sigma_T(0)\gamma^0}\frac{\mathrm{d}}{\mathrm{d}x}\psi_0(0) + \mathcal{O}(\epsilon^2). \tag{3.69}$$

Habetler and Matkowsky [34] use formulae from Case and Zweifel [20] to evaluate γ^0 and γ^1. The coefficient γ^1 is also found by Jin and Levermore [40] to a higher accuracy. These values are

$$\begin{aligned}\gamma^0 &= 1, \\ \gamma^1 &\approx 0.710446,\end{aligned}$$

(see [34], equations (4.43) and (4.44) and [40], equation (4.1)). We can now conclude that, to $\mathcal{O}(\epsilon^2)$, (3.69) is equivalent to the condition

$$\phi_{[1]}(0) - \epsilon\frac{\gamma^1}{\sigma_T(0)}\frac{\mathrm{d}}{\mathrm{d}x}\phi_{[1]}(0) = \int_{[0,1]} f_L(\mu)\gamma(\mu)\,\mathrm{d}\mu. \tag{3.70}$$

Equation (3.70) is the boundary condition for the lower end of the spatial domain that

we were aiming for. Note that if we were imposing vacuum boundary conditions on the transport equation (3.4), then $f_L = f_R = 0$ and so the right hand side would just be zero.

An equivalent inner asymptotic expansion in the boundary layer at the upper end of the spatial domain (near $x = 1$, using the scaling $x = 1 - \epsilon y$) and appropriate matching procedure can be carried out in a similar way to our above work. Up to $\mathcal{O}(\epsilon^2)$ this results in

$$\phi_{[1]}(1) + \epsilon \frac{\gamma^1}{\sigma_T(1)} \frac{\mathrm{d}}{\mathrm{d}x} \phi_{[1]}(1) = \int_{[-1,0]} f_R(\mu)\gamma(\mu)\, \mathrm{d}\mu. \tag{3.71}$$

3.2.6 Summary

We have found (through the outer expansion in Section 3.2.2) a diffusion equation that approximates the scalar flux to an error of $\mathcal{O}(\epsilon^2)$. We have also obtained (through the inner expansion and subsequent matching procedure in Sections 3.2.3, 3.2.4 and 3.2.5) suitable boundary conditions for this equation at each end of the domain. This diffusion equation can provide a useful approximation under appropriate conditions, and such conditions could be found by comparing the approximate scalar flux with the flux from the transport equation. However in the next section we will see that the diffusion approximation can be used to *accelerate* the source iteration method. In particular we will note that in practice, for all tested material properties, this acceleration improves upon the rate of convergence of source iteration. Consequently the question of whether to utilize the diffusion approximation comes down to the computational cost of doing so, and this question is tackled by Adams and Larsen [2, Section II.B]. The importance of such a consideration will become clear when applying localised diffusion acceleration as we do in Section 4.5.3.

We will conclude this section by summarising the obtained results in both the dimensionless and dimensional forms. We will return to using the 'hat' notation ($\hat{\cdot}$) to denote dimensionless quantities, as in Section 3.2.1.

Firstly, equations (3.31), (3.70) and (3.71) are the diffusion equation and associated boundary conditions. We restate these here, using the earlier nondimensional notation, as

$$-\frac{\mathrm{d}}{\mathrm{d}\hat{x}} \left(\frac{1}{3\hat{\sigma}_T(\hat{x})} \frac{\mathrm{d}}{\mathrm{d}\hat{x}} \hat{\phi}_{[1]}(\hat{x}) \right) + \hat{\sigma}_A(\hat{x}) \hat{\phi}_{[1]}(\hat{x}) = \hat{\sigma}_T(\hat{x}) \frac{\hat{Q}(\hat{x})}{\epsilon^2}, \tag{3.72}$$

subject to boundary conditions

$$\hat{\phi}_{[1]}(0) - \epsilon \frac{\gamma^1}{\hat{\sigma}_T(0)} \frac{\mathrm{d}}{\mathrm{d}\hat{x}} \hat{\phi}_{[1]}(0) = \int_{[0,1]} \hat{f}_L(\mu)\gamma(\mu)\, \mathrm{d}\mu, \tag{3.73}$$

$$\hat{\phi}_{[1]}(1) + \epsilon \frac{\gamma^1}{\hat{\sigma}_T(1)} \frac{\mathrm{d}}{\mathrm{d}\hat{x}} \hat{\phi}_{[1]}(1) = \int_{[-1,0]} \hat{f}_R(\mu)\gamma(\mu)\, \mathrm{d}\mu. \tag{3.74}$$

Next we would like to write down the dimensional form of this system. To do this we must recall the scalings that were taken in Section 3.2.1, namely

$$x = D\hat{x}, \quad \psi(x,\mu) = \langle\psi\rangle\hat{\psi}(\hat{x},\mu), \quad Q(x) = \langle\psi\rangle\sigma_T(x)\hat{Q}(\hat{x}),$$
$$\sigma_T(x) = \langle\sigma_T\rangle\hat{\sigma}_T(\hat{x}), \quad c(x) = \hat{c}(\hat{x}),$$

as well as the asymptotic variable given in (3.2), the σ_A scaling from (3.29) and the boundary condition scalings, which were respectively

$$\epsilon \equiv \frac{1}{\langle\sigma_T\rangle D},$$
$$D\sigma_A(x) = \epsilon\hat{\sigma}_A(\hat{x}),$$
$$f_L(\mu) \equiv \langle\psi\rangle\hat{f}_L,$$
$$f_R(\mu) \equiv \langle\psi\rangle\hat{f}_R.$$

With these scalings, the nondimensional diffusion equation is equal to

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{3\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\phi_{[1]}(x)\right) + \sigma_A(x)\phi_{[1]}(x) = Q(x), \tag{3.75}$$

in which,

$$\phi_{[1]}(x) = \langle\psi\rangle\hat{\phi}_{[1]}(\hat{x}),$$

is a dimensional approximation to the scalar flux. Also, the boundary conditions become

$$\phi_{[1]}(0) - \frac{\gamma^1}{\sigma_T(0)}\frac{\mathrm{d}}{\mathrm{d}x}\phi_{[1]}(0) = \int_{[0,1]} f_L(\mu)\gamma(\mu)\, \mathrm{d}\mu, \tag{3.76}$$

$$\phi_{[1]}(D) + \frac{\gamma^1}{\sigma_T(D)}\frac{\mathrm{d}}{\mathrm{d}x}\phi_{[1]}(D) = \int_{[-1,0]} f_R(\mu)\gamma(\mu)\, \mathrm{d}\mu. \tag{3.77}$$

This dimensional version of the diffusion equation ties in with the notation used else-

where in this thesis.

3.3 Diffusion Synthetic Acceleration

In Chapter 2 we proved that source iteration applied to a simplified version of the neutron transport equation converges provided the uniform norm of the scattering ratio, $\|\sigma_S/\sigma_T\|_\infty$, is less than one (see Theorem 2.21). However we will also see in Section 3.5.1 that the rate of convergence of source iteration is potentially impractically poor when this norm is close to one. In this section we describe a well known method called *diffusion synthetic acceleration* (DSA) that has the potential to overcome this slow down. We start in Section 3.3.1 by understanding what a general *synthetic acceleration* method entails. Then in Section 3.3.2 we use the diffusion approximation derived in Section 3.2 to obtain the method known as DSA in Algorithm 3. Lastly, in Section 3.3.3 we describe how DSA can be understood as a preconditioner to source iteration.

3.3.1 Synthetic Acceleration

Synthetic acceleration schemes were originally introduced by Kopp in 1963 [48]. The idea of such schemes is to improve the convergence of a basic iterative method by 'updating' the current approximate solution at each iteration. Consequently they can be thought of as two-step methods: the first step is one iteration of the basic iterative method; the second step is an update to the approximation found in step one. Ideally the method used to obtain the update should complement the original method, causing the accelerated method to converge rapidly over a wider range of situations than the basic method.

We will demonstrate this by building a synthetic acceleration scheme based upon the source iteration algorithm that was the subject of Chapter 2, and will follow the method as outlined in [2, Section I.E.]. Within this subsection we will work in the 3D setting defined in Section 2.2.1, and as such will use the equations, operators and notation introduced there.

We start by rewriting (2.52), the equation for the error at the $(k+1)$th step of source iteration, as

$$\phi(\mathbf{r}) - \phi^{(k+1/2)}(\mathbf{r}) = \mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r})\left(\phi(\mathbf{r}) - \phi^{(k)}(\mathbf{r})\right),$$

where, instead of the $(k+1)$th iterate, we are now iterating to find a half-step approximation, denoted $\phi^{(k+1/2)}$. Adding and subtracting $\phi^{(k+1/2)}$ to the brackets on the right, expanding and rearranging leaves

$$(\mathcal{I} - \mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r}))\left(\phi(\mathbf{r}) - \phi^{(k+1/2)}(\mathbf{r})\right) = \mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r})\left(\phi^{(k+1/2)}(\mathbf{r}) - \phi^{(k)}(\mathbf{r})\right). \quad (3.78)$$

We now have an equation for the correction, $\phi(\mathbf{r}) - \phi^{(k+1/2)}(\mathbf{r})$, via the solution

$$\phi(\mathbf{r}) - \phi^{(k+1/2)}(\mathbf{r}) = (\mathcal{I} - \mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r}))^{-1}\mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r})\left(\phi^{(k+1/2)}(\mathbf{r}) - \phi^{(k)}(\mathbf{r})\right), \quad (3.79)$$

which (as the difference between two functions with equal boundary values) is subject to zero boundary conditions. However solving for this correction is just as hard as solving the original transport equation. Instead we introduce a new quantity

$$\delta(\mathbf{r}) = \mathcal{M}\mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r})\left(\phi^{(k+1/2)}(\mathbf{r}) - \phi^{(k)}(\mathbf{r})\right), \quad (3.80)$$

whereby if $\mathcal{M} \approx (\mathcal{I} - \mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r}))^{-1}$ then $\delta(\mathbf{r}) \approx \phi(\mathbf{r}) - \phi^{(k+1/2)}(\mathbf{r})$. The operator $\mathcal{M}$ should be easy to calculate, but still be a good enough approximate inverse that we can use $\delta(\mathbf{r})$ to improve upon the half-step approximation to the flux via

$$\phi^{(k+1)}(\mathbf{r}) = \phi^{(k+1/2)}(\mathbf{r}) + \delta(\mathbf{r}). \quad (3.81)$$

Before we write down Algorithm 2, we first confirm some notation. In Section 2.2 we introduced two operators, $\mathcal{T}$ and $\mathcal{P}$, in 1,2 and 3D in order to simplify the expression of the dimensional transport equation. We remark here that we will follow the notational style of Keener, [46, p.151], whereby the symbol $\mathcal{T}$ includes both the differential operator as well as the domain with given boundary conditions. Therefore we can use the inverse operator, $\mathcal{T}^{-1}$, to denote the action of solving the transport equation subject to the included boundary conditions, as was completed in Lemma 2.2. This allows one step of source iteration (Algorithm 1) to be written succinctly as

$$\phi^{(k+1)}(\mathbf{r}) = \mathcal{P}\mathcal{T}^{-1}\left(\sigma_S(\mathbf{r})\phi^{(k)}(\mathbf{r}) + Q(\mathbf{r})\right).$$

We can use this convention, along with the above framework, and write down Algorithm 2: a general synthetic accelerated source iteration algorithm aimed at solving the transport equation given by (2.1), subject to boundary conditions (2.2).

Algorithm 2: Synthetic Accelerated Source Iteration

1. Start with some initial $\phi^{(0)}(\mathbf{r})$.

2. Find $\phi^{(k+1/2)}(\mathbf{r})$ that satisfies

$$\phi^{(k+1/2)}(\mathbf{r}) = \mathcal{P}\mathcal{T}^{-1}\left(\sigma_S(\mathbf{r})\phi^{(k)}(\mathbf{r}) + Q(\mathbf{r})\right), \tag{3.82}$$

   subject to boundary conditions (2.2).

3. Find $\delta^{(k+1/2)}(\mathbf{r})$ that satisfies

$$\delta^{(k+1/2)}(\mathbf{r}) = \mathcal{M}\mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r})\left(\phi^{(k+1/2)}(\mathbf{r}) - \phi^{(k)}(\mathbf{r})\right), \tag{3.83}$$

   where $\mathcal{M} \approx (\mathcal{I} - \mathcal{K}_{\sigma_T}\sigma_S(\mathbf{r}))^{-1}$.

4. Update the scalar flux approximation to find

$$\phi^{(k+1)}(\mathbf{r}) = \phi^{(k+1/2)}(\mathbf{r}) + \delta^{(k+1/2)}(\mathbf{r}), \tag{3.84}$$

   and return to step 2.

The effectiveness of such a method hangs upon the choice of the approximate solution operator, $\mathcal{M}$. It would be sensible to find an approximation that is most accurate in situations where source iteration performs poorly, i.e. when $\|\sigma_S/\sigma_T\|_\infty$ is close to one. This is a criteria satisfied by the diffusion approximation derived in Section 3.2, and using the diffusion approximation as an approximate solution operator results in the method called diffusion synthetic acceleration.

We conclude this section by mentioning that by writing source iteration as a simple *Richardson* iteration (L. F. Richardson, 1911, [62]), it can be shown that synthetic accelerated source iteration is equivalent to a preconditioned Richardson iteration scheme [35]. Before showing this we first recall that for a model problem, $\mathcal{A}x = b$, Richardson iteration is obtained by rewriting as $0 = b - \mathcal{A}x$, and then adding x to both sides. As an iterative scheme this then yields

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \left(b - \mathcal{A}x^{(k)}\right) \\ &= \left(\mathcal{I} - \mathcal{A}\right) x^{(k)} + b. \end{aligned}$$

To obtain the preconditioned Richardson scheme we multiply by a preconditioner, say $\mathbb{P}$, before adding x to both sides. Therefore preconditioned Richardson iteration is given by

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \mathbb{P}\left(b - \mathcal{A}x^{(k)}\right) \\ &= \left(\mathcal{I} - \mathbb{P}\mathcal{A}\right) x^{(k)} + \mathbb{P}b. \end{aligned} \tag{3.85}$$

Now, applying the operators $\mathcal{P}$ and $\mathcal{T}^{-1}$ to the simplified neutron transport equation (2.5) with non-zero boundary conditions (2.2) we obtain an equation for the scalar flux,

$$\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)\phi = \mathcal{P}\mathcal{T}^{-1}Q. \tag{3.86}$$

For this the Richardson iteration scheme is given by

$$\begin{aligned} \phi^{(k+1)} &= \left(\mathcal{I} - \left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)\right)\phi^{(k)} + \mathcal{P}\mathcal{T}^{-1}Q \\ &= \mathcal{P}\mathcal{T}^{-1}\sigma_S\phi^{(k)} + \mathcal{P}\mathcal{T}^{-1}Q, \end{aligned} \tag{3.87}$$

which is equivalent to the source iteration algorithm presented in Chapter 2. To show that synthetic accelerated source iteration is the preconditioned form of (3.87) we follow the work of Adams and Larsen, [2, Section I.E.] and recall equation (3.79), which we can write as

$$\phi^{(k+1)} = \phi^{(k+1/2)} + \mathcal{M}\mathcal{P}\mathcal{T}^{-1}\sigma_S\left(\phi^{(k+1/2)} - \phi^{(k)}\right).$$

Eliminating $\phi^{(k+1/2)}$ by substituting in (3.82) results in

$$\phi^{(k+1)} = \mathcal{P}\mathcal{T}^{-1}\sigma_S\phi^{(k)} + \mathcal{P}\mathcal{T}^{-1}Q + \mathcal{M}\mathcal{P}\mathcal{T}^{-1}\sigma_S\left(\mathcal{P}\mathcal{T}^{-1}\sigma_S\phi^{(k)} + \mathcal{P}\mathcal{T}^{-1}Q - \phi^{(k)}\right). \tag{3.88}$$

After some rearrangement, this becomes

$$\phi^{(k+1)} = \phi^{(k)} - \left(\mathcal{I} + \mathcal{M}\mathcal{P}\mathcal{T}^{-1}\sigma_S\right)\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)\phi^{(k)} + \left(\mathcal{I} + \mathcal{M}\mathcal{P}\mathcal{T}^{-1}\sigma_S\right)\mathcal{P}\mathcal{T}^{-1}Q,$$

and so defining $\mathbb{P} = \mathcal{I} + \mathcal{M}\mathcal{P}\mathcal{T}^{-1}\sigma_S$ we are left with

$$\phi^{(k+1)} = \left(\mathcal{I} - \mathbb{P}\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)\right)\phi^{(k)} + \mathbb{P}\mathcal{P}\mathcal{T}^{-1}Q. \tag{3.89}$$

This is the preconditioned form of (3.87) with preconditioner $\mathbb{P}$, and comparison with (3.85) shows it is a preconditioned Richardson scheme. If we suppose that $\mathcal{M} \approx \left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S(\mathbf{r})\right)^{-1}$ then

$$\begin{aligned}\mathbb{P} = \mathcal{I} + \mathcal{M}\mathcal{P}\mathcal{T}^{-1}\sigma_S &\approx \mathcal{I} + \left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)^{-1}\mathcal{P}\mathcal{T}^{-1}\sigma_S \\ &= \left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)^{-1}\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S + \mathcal{P}\mathcal{T}^{-1}\sigma_S\right),\end{aligned}$$

leaving us with

$$\mathbb{P} \approx \left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)^{-1},$$

and so (3.89) might be reasonably expected to converge quickly. For further details see Adams and Larsen, [2, Section I.E.] or Warsa et. al., [75, Section III.B].

3.3.2 DSA Algorithm

In this section we will combine the synthetic acceleration algorithm (Algorithm 2) with the diffusion approximation derived in Section 3.2. The result will be the well-known algorithm called *diffusion synthetic acceleration*, or DSA, which uses the diffusion approximation (3.75) as the solution operator, $\mathcal{M}$. This name was first used by R. E. Alcouffe in 1976 [4][5], where it was shown that DSA is rapidly convergent over all spatial mesh sizes, overcoming a problem highlighted earlier by W. H. Reed, [60]. A very good history of the development of, and subsequent research on, DSA can be found in Adams and Larsen, [2, Section I.G, p.16]. Later on Adams and Larsen also weigh the extra computational expense of DSA against its improved convergence rate, using a Fourier argument to derive conditions for which DSA is more economical than source iteration (see [2, Section II.B]).

We will work in the 1D slab geometry specified in Section 2.2.3, with spatial domain $[x_L, x_R]$ and angular domain $[-1, 1]$. Applying the operators $\mathcal{P}$ and $\mathcal{T}^{-1}$ to the simplified neutron transport equation (2.13) with non-zero boundary conditions (2.14)

we obtain an equation for the 1D scalar flux,

$$\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S(x)\right)\phi(x) = \mathcal{P}\mathcal{T}^{-1}Q(x). \tag{3.90}$$

We also know that (in diffusive regimes, as explained in Section 3.2.1) the solution, ϕ, can be approximated by the solution to the diffusion equation (3.75), which we restate here for convenience

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{3\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\Theta(x)\right) + \sigma_A(x)\Theta(x) = Q(x), \tag{3.91}$$

subject to boundary conditions

$$\Theta(x_L) - \frac{\gamma^1}{\sigma_T(x_L)}\frac{\mathrm{d}}{\mathrm{d}x}\Theta(x_L) = \int_{[0,1]} f_L(\mu)\gamma(\mu)\,\mathrm{d}\mu, \tag{3.92}$$

$$\Theta(x_R) + \frac{\gamma^1}{\sigma_T(x_R)}\frac{\mathrm{d}}{\mathrm{d}x}\Theta(x_R) = \int_{[-1,0]} f_R(\mu)\gamma(\mu)\,\mathrm{d}\mu, \tag{3.93}$$

with $\gamma^1 \approx 0.710446$ and where we know $\phi = \Theta + O(\epsilon^2)$ in which ϵ is an asymptotic variable. To simplify this notation we will introduce the 1-dimensional differential operator $\mathcal{D} : L^2[x_L, x_R] \to L^2[x_L, x_R]$, defined as follows

$$\mathcal{D}u(x) \equiv -\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{3\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}u(x)\right) + \sigma_A(x)u(x), \tag{3.94}$$

with u satisfying the boundary conditions (3.92) and (3.93). For convenience we will again follow the notational style of Keener, [46, p.151], and allow the symbol $\mathcal{D}$ to include both the formal differential operation as well as the domain and given boundary conditions. With this notation the above diffusion problem (3.91)-(3.93) can be written simply as

$$\mathcal{D}\Theta(x) = Q(x).$$

Remark 3.2:
The diffusion operator, $\mathcal{D}$, is a Sturm-Liouville *operator, with corresponding regular Sturm-Liouville problem*

$$\mathcal{D}u(x) = \lambda u(x).$$

Since in (3.94) $\sigma_A > 0$*, the operator $\mathcal{D}$ is positive-definite (as can be verified by taking the inner product $\langle \mathcal{D}u, u\rangle$ and integrating by parts, see Keener, [46, p.163]). In particular this implies that*

$$\langle \mathcal{D}u, u \rangle = 0 \Leftrightarrow u = 0.$$

Consequently by the Fredholm Alternative Theorem (see Keener, [46, Section 4.3, Theorem 4.4]) we know that for any $f \in L^2[a,b]$ the system $\mathcal{D}u(x) = f(x)$ has a unique solution, and so we can say that this system has a solution given by

$$u(x) = \mathcal{D}^{-1} f(x). \tag{3.95}$$

This inverse operator will be useful later in Section 3.3.3.

Algorithm 3: Diffusion Synthetic Acceleration

1. Start with some initial $\phi^{(0)}(x)$.

2. Find $\phi^{(k+1/2)}(x)$ that satisfies

$$\phi^{(k+1/2)}(x) = \mathcal{P}\mathcal{T}^{-1}\left(\sigma_S(x)\phi^{(k)}(x) + Q(x)\right), \tag{3.96}$$

   subject to boundary conditions (2.14).

3. Find $\delta^{(k+1/2)}(x)$ that satisfies

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{3\sigma_T(x)}\frac{\mathrm{d}}{\mathrm{d}x}\delta^{(k+1/2)}(x)\right) + \sigma_A(x)\delta^{(k+1/2)}(x) = \sigma_S(x)\left(\phi^{(k+1/2)}(x) - \phi^{(k)}(x)\right), \tag{3.97}$$

   subject to

$$\delta^{(k+1/2)}(x_L) - \frac{\gamma^1}{\sigma_T(x_L)}\frac{\mathrm{d}}{\mathrm{d}x}\delta^{(k+1/2)}(x_L) = 0, \tag{3.98}$$

$$\delta^{(k+1/2)}(x_R) + \frac{\gamma^1}{\sigma_T(x_R)}\frac{\mathrm{d}}{\mathrm{d}x}\delta^{(k+1/2)}(x_R) = 0. \tag{3.99}$$

4. Update the scalar flux approximation to find

$$\phi^{(k+1)}(x) = \phi^{(k+1/2)}(x) + \delta^{(k+1/2)}(x), \tag{3.100}$$

   and return to step 2.

The crucial observation is that the diffusion equation can be used to approximate the solution to (3.78), and so can provide the required approximate solution operator, $\mathcal{M}$, in Algorithm 2. By applying the diffusion equation in this way, we find Algorithm 3 for diffusion synthetic acceleration.

The DSA algorithm is also presented in this two-step form in [52].

3.3.3 DSA as a Preconditioner

Algorithm 3 presents DSA as a two-step iterative method. We will now describe how DSA can be understood as a preconditioning to source iteration, as was initially explored by Faber and Manteuffel in [27] in a 1D setting. Later in 1991, Ashby et. al. [6] built on the work of Larsen [51] and found a preconditioner in a discrete setting that was equivalent to that of Faber and Manteuffel. In 1995 Ashby et. al. [7] gave a clearer statement of this preconditioner (with the same work for a different discretisation completed in [15]). The equivalence of the two preconditioners can be seen in (3.106).

Earlier we mentioned that two-step synthetic accelerated source iteration (Algorithm 2) is equivalent to preconditioned Richardson iteration, given by (3.89) (see [2, Section I.E] for more detail). In Lemma 3.3 we will confirm that DSA can be written in this way by using the preconditioner found in Ashby et. al. [7].

Lemma 3.3:
Algorithm 3 is equivalent to the preconditioned Richardson scheme,

$$\phi^{(k+1)}(x) = \left(\mathcal{I} - \mathbb{P}\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S(x)\right)\right)\phi^{(k)}(x) + \mathbb{P}\mathcal{P}\mathcal{T}^{-1}Q(x), \tag{3.101}$$

where

$$\mathbb{P} \equiv \mathcal{I} + \mathcal{D}^{-1}\sigma_S(x), \tag{3.102}$$

with $\mathcal{D} : L^2[x_L, x_R] \to L^2[x_L, x_R]$ *defined in* (3.94).

Proof.
In Algorithm 3 we solve two equations, (3.96) and (3.97), subject to boundary conditions (3.98) and (3.99). We restate these two equations here (dropping the spatial dependencies and including (3.84) also) as

$$\phi^{(k+1/2)} = \mathcal{P}\mathcal{T}^{-1}\left(\sigma_S\phi^{(k)} + Q\right), \tag{3.103}$$

$$\mathcal{D}\left(\phi^{(k+1)} - \phi^{(k+1/2)}\right) = \sigma_S\left(\phi^{(k+1/2)} - \phi^{(k)}\right). \tag{3.104}$$

Since we know $\mathcal{D}$ is invertible (see Remark 3.2) we can solve (3.104) to obtain

$$\begin{aligned}\phi^{(k+1)} - \phi^{(k+1/2)} &= \mathcal{D}^{-1}\sigma_S\left(\phi^{(k+1/2)} - \phi^{(k)}\right) \\ &= \mathcal{D}^{-1}\sigma_S\phi^{(k+1/2)} - \mathcal{D}^{-1}\sigma_S\phi^{(k)},\end{aligned}$$

whereby combining terms involving $\phi^{(k+1/2)}$, and adding and subtracting $\phi^{(k)}$, we have

$$\phi^{(k+1)} = \phi^{(k)} - \phi^{(k)} - \mathcal{D}^{-1}\sigma_S\phi^{(k)} + \left(\mathcal{I} + \mathcal{D}^{-1}\sigma_S\right)\phi^{(k+1/2)}.$$

Next using (3.103) and combining two terms involving $\phi^{(k)}$ we find

$$\phi^{(k+1)} = \phi^{(k)} - \left(\mathcal{I} + \mathcal{D}^{-1}\sigma_S\right)\phi^{(k)} + \left(\mathcal{I} + \mathcal{D}^{-1}\sigma_S\right)\left(\mathcal{P}\mathcal{T}^{-1}\sigma_S\phi^{(k)} + \mathcal{P}\mathcal{T}^{-1}Q\right).$$

so that expanding and combining like terms leaves us with

$$\phi^{(k+1)} = \phi^{(k)} - \left(\mathcal{I} + \mathcal{D}^{-1}\sigma_S\right)\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)\phi^{(k)} + \left(\mathcal{I} + \mathcal{D}^{-1}\sigma_S\right)\mathcal{P}\mathcal{T}^{-1}Q.$$

Lastly, using the preconditioner definition (3.102) results in (3.101) as required. □

As outlined in Adams and Larsen [2, Section I.E], this preconditioned Richardson scheme should converge quickly provided $\mathbb{P}\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S(x)\right) \approx \mathcal{I}$. We will return to this condition in the discussion at the end of Section 3.4.

As mentioned, a discrete form this preconditioner is given by Ashby et. al. in [7]. They give a preconditioned source iteration algorithm in Section 4, p. 143, with the preconditioner defined in a discrete block-matrix setting in Section 5, equation (5.6).

Faber and Manteuffel [27] describe a different approach to finding a preconditioned form of DSA. After some manipulation, the preconditioner they derive can be written down as

$$\mathbb{P}_{FM} \equiv \left(\mathcal{W} - \frac{\sigma_S(x)}{\sigma_T(x)}\mathcal{I}\right)^{-1}\mathcal{W}, \tag{3.105}$$

where

$$\mathcal{W} \equiv \frac{1}{\sigma_T(x)}\mathcal{D}\left(\mathcal{I} + \mathcal{D}^{-1}\sigma_S(x)\right).$$

As mentioned before, this is equivalent to the preconditioner found later in [7], and to see this we once again drop the spatial dependencies and equate

$$\left(\mathcal{W}-\frac{\sigma_S}{\sigma_T}\mathcal{I}\right)^{-1}\mathcal{W}=\left(\frac{1}{\sigma_T}\mathcal{D}\left(\mathcal{I}+\mathcal{D}^{-1}\sigma_S\right)-\frac{\sigma_S}{\sigma_T}\mathcal{I}\right)^{-1}\frac{1}{\sigma_T}\mathcal{D}\left(\mathcal{I}+\mathcal{D}^{-1}\sigma_S\right),$$
$$=\left(\left(\mathcal{D}+\sigma_S\mathcal{I}\right)-\sigma_S\mathcal{I}\right)^{-1}\sigma_T\frac{1}{\sigma_T}\mathcal{D}\left(\mathcal{I}+\mathcal{D}^{-1}\sigma_S\right),$$
$$=\mathcal{D}^{-1}\mathcal{D}\left(\mathcal{I}+\mathcal{D}^{-1}\sigma_S\right)=\left(\mathcal{I}+\mathcal{D}^{-1}\sigma_S\right). \tag{3.106}$$

In their paper Faber and Manteuffel apply DSA as a preconditioner not only to source iteration, but also to the conjugate gradient (CG) method (see [66]). They find the preconditioned CG implementation of DSA to have a considerably faster rate of convergence.

In [9] and [10] it was first understood that discontinuities in material properties can severely reduce the effectiveness of DSA. Ashby et. al., [6], and later Warsa et. al., [74],[75], showed that using DSA as a preconditioner to Krylov methods was an effective way to overcome this issue as well as having other advantages over preconditioned source iteration, though we do not explore this idea any further here.

In this thesis we will only implement DSA as an accelerated source iteration algorithm, however the application of DSA as a preconditioner to Krylov methods is very powerful. In domains with highly discontinuous material properties it has certainly been seen to converge much faster than diffusion accelerated source iteration [75]. On the other hand in domains with homogeneous material properties diffusion accelerated source iteration also converges quickly and is computationally cheaper to execute. Furthermore, away from diffusive regions it may not be necessary to implement any form of DSA since more basic methods also exhibit fast convergence (see Section 2.7). These domain-dependent requirements motivated us to consider domain decomposition methods that would allow the most appropriate iterative method to be applied in different areas of the spatial domain. This work is presented in Chapter 4 where two different domain decomposition methods are defined, analysed and implemented numerically.

3.4 Block Operator Diffusion

We have seen, using an asymptotic expansion argument, that under certain conditions the scalar flux can be well approximated by a diffusion equation of the form (3.72). In this section we will use a block operator argument to show the link between the diffusion and transport equations under the assumption of zero boundary conditions. Specifically we will show that, under certain assumptions, a scaled Schur complement equation ((3.114) below, arising from a block operator form of the transport equation)

is an $\mathcal{O}(\epsilon)$ approximation of the diffusion equation, (3.72). To do this we will need to use the nondimensional notation established in Section 3.2.

First we recall the nondimensional transport equation given in (3.1),

$$\frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\hat{\psi}(\hat{x},\mu)+\hat{\psi}(\hat{x},\mu)=\frac{\hat{c}(\hat{x})}{2}\int_{[-1,1]}\hat{\psi}(\hat{x},\mu)\,\mathrm{d}\mu+\hat{Q}(\hat{x}),$$

in which

$$x=D\hat{x},\quad \psi(x,\mu)=\langle\psi\rangle\hat{\psi}(\hat{x},\mu),\quad Q(x)=\langle\psi\rangle\sigma_T(x)\hat{Q}(\hat{x}),$$
$$D\sigma_T(x)=\hat{\sigma}_T(\hat{x})/\epsilon,\quad c(x)=\hat{c}(\hat{x}),\quad D\sigma_A(x)=\epsilon\hat{\sigma}_A(\hat{x}),$$

where $\hat{x}\in[0,1]$ is dimensionless, $\mu\in[-1,1]$, and $\epsilon=1/(\langle\sigma_T\rangle D)$. To this we apply zero boundary conditions, (3.3) with $\hat{f}_L(\mu)=\hat{f}_L(\mu)=0$, which we restate here

$$\hat{\psi}(0,\mu)=0,\quad \text{when } \mu>0,$$
$$\hat{\psi}(1,\mu)=0,\quad \text{when } \mu<0.$$

We also note that during the outer asymptotic expansion (Section 3.2.2) it was shown that, to have a meaningful, non-trivial solution, we require

$$\hat{Q}(\hat{x})=\epsilon^2\hat{Q}_2(\hat{x})+\mathcal{O}(\epsilon^3), \tag{3.107}$$

$$\hat{c}(\hat{x})=1-\epsilon^2\frac{\hat{\sigma}_A(\hat{x})}{\hat{\sigma}_T(\hat{x})}+\mathcal{O}(\epsilon^3). \tag{3.108}$$

In Section 2.2.3 we defined two operators, $\mathcal{T}$ and $\mathcal{P}$, in (2.15) and (2.17) respectively. We recall here that we are using the notational style of Keener, [46, p.151], and so the symbol $\mathcal{T}$ includes the differential operation as well as the domain and boundary conditions. We now define the dimensionless operator

$$\hat{\mathcal{T}}\equiv\frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}+\mathcal{I}, \tag{3.109}$$

where $\mathcal{I}$ is the identity operator, and $\hat{\mathcal{T}}$ also imposes the dimensionless boundary conditions (3.3).

Remark 3.4:
Note that the operators $\mathcal{T}$ and $\hat{\mathcal{T}}$ are related by

$$\hat{\mathcal{T}}(\cdot)\equiv\frac{\epsilon D}{\hat{\sigma}_T(\hat{x})}\mathcal{T}(\cdot).$$

From this we can see that

$$\hat{\mathcal{T}}^{-1}(\cdot) \equiv \mathcal{T}^{-1}\left(\frac{\hat{\sigma}_T(\hat{x})}{\epsilon D}(\cdot)\right), \tag{3.110}$$

acts as an inverse for $\hat{\mathcal{T}}$ *(noting that it contains appropriate boundary conditions), i.e.* $w(\hat{x},\mu) \equiv \hat{\mathcal{T}}^{-1}g(\hat{x},\mu)$ *solves*

$$\hat{\mathcal{T}}w(\hat{x},\mu) = g(\hat{x},\mu),$$

and also satisfies the nondimensional boundary conditions given by (3.3)*, for* $w, g \in L^2([0,1], L^\infty[-1,1])$*. Therefore, by taking* (2.28) *as the definition of* $\mathcal{T}^{-1}$*, we can see that for any* $f \in L^\infty[-1,1]$ *and any* $h \in L^2[0,1]$*,* $\hat{\mathcal{T}}^{-1}$ *satisfies*

$$\hat{\mathcal{T}}^{-1}(f(\mu)h(\hat{x})) = f(\mu)\hat{\mathcal{T}}^{-1}h(\hat{x}). \tag{3.111}$$

Using the operator $\hat{\mathcal{T}}$ we can write the dimensionless neutron transport equation in operator form as

$$\hat{\mathcal{T}}\hat{\psi}(\hat{x},\mu) = \hat{c}(\hat{x})\hat{\phi}(\hat{x}) + \hat{Q}(\hat{x}),$$

with

$$\hat{\phi}(\hat{x}) = \mathcal{P}\hat{\psi}(\hat{x},\mu).$$

We can combine these two equations into a block operator form as follows

$$\begin{pmatrix} \hat{\mathcal{T}} & -\hat{c}\mathcal{I} \\ -\mathcal{P} & \mathcal{I} \end{pmatrix}\begin{pmatrix} \hat{\psi} \\ \hat{\phi} \end{pmatrix} = \begin{pmatrix} \hat{Q} \\ 0 \end{pmatrix}. \tag{3.112}$$

Applying a Gaussian elimination type process to (3.112) we pre-multiply as follows

$$\begin{pmatrix} \mathcal{I} & 0 \\ \mathcal{P}\hat{\mathcal{T}}^{-1} & \mathcal{I} \end{pmatrix}\begin{pmatrix} \hat{\mathcal{T}} & -\hat{c}\mathcal{I} \\ -\mathcal{P} & \mathcal{I} \end{pmatrix}\begin{pmatrix} \hat{\psi} \\ \hat{\phi} \end{pmatrix} = \begin{pmatrix} \mathcal{I} & 0 \\ \mathcal{P}\hat{\mathcal{T}}^{-1} & \mathcal{I} \end{pmatrix}\begin{pmatrix} \hat{Q} \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} \hat{\mathcal{T}} & -\hat{c}\mathcal{I} \\ 0 & \mathcal{I} - \mathcal{P}\hat{\mathcal{T}}^{-1}\hat{c} \end{pmatrix}\begin{pmatrix} \hat{\psi} \\ \hat{\phi} \end{pmatrix} = \begin{pmatrix} \hat{Q} \\ \mathcal{P}\hat{\mathcal{T}}^{-1}\hat{Q} \end{pmatrix}. \tag{3.113}$$

Here we have obtained a *Schur complement* equation for the scalar flux, which for later convenience we will scale by $\epsilon^{-2}\hat{\sigma}_T(\hat{x})$, and so can be written as

$$\frac{\hat{\sigma}_T(\hat{x})}{\epsilon^2}(\mathcal{I} - \mathcal{P}\hat{\mathcal{T}}^{-1}\hat{c})\hat{\phi} = \frac{\hat{\sigma}_T(\hat{x})}{\epsilon^2}\mathcal{P}\hat{\mathcal{T}}^{-1}\hat{Q}. \tag{3.114}$$

Inverting this Schur complement operator is as difficult as solving the original transport equation. Instead, we will show in Theorem 3.5 that the Schur complement can be approximated by the diffusion equation found in Section 3.2. Furthermore, in Corollary 3.10 we will show how in the limit as ϵ tends to zero, the right hand side of (3.114) tends to just a scaling of the source term. Together these results will show a link between the Schur complement equation, (3.114), and the diffusion equation (3.72) derived in Section 3.2. This link is a new interpretation of the relationship between the transport equation and diffusion equation, demonstrated asymptotically in Section 3.2, and we consider the potential implications of this new link at the end of this section.

We start by proving Theorem 3.5, which relates the Schur operator on the left side of (3.114) to the diffusion equation, (3.31). In the proof of Theorem 3.5 we will make use of the fact that for any operator $\mathcal{A}$, if $(\mathcal{I}+\mathcal{A})^{-1}$ exists, then

$$(\mathcal{I}+\mathcal{A})^{-1} = \mathcal{I} - \mathcal{A} + \mathcal{A}^2 - \mathcal{A}^3 + (\mathcal{I}+\mathcal{A})^{-1}\mathcal{A}^4. \tag{3.115}$$

Therefore, for any function, f, a bound on $(\mathcal{I}+\mathcal{A})^{-1}f$ can be found by bounding the last term of the expansion, $(\mathcal{I}+\mathcal{A})^{-1}\mathcal{A}^4 f$ provided f is smooth enough. Finding such a bound will occupy us for the second half of the proof of Theorem 3.5.

In the statement of this proof we refer to the Sobolev space, $W^{m,p}[0,1]$, which is defined as

$$W^{m,p}[0,1] \equiv \{u \in L^p[0,1] : \partial^\alpha u \in L^p[0,1] \quad \forall \alpha \in \mathbb{N}_0 \text{ st. } |\alpha| \leq m\}$$

where ∂^α represents the derivative in a weak sense. We also use the related Hilbert space, which can be defined as $H^m[0,1] \equiv W^{m,2}[0,1]$.

Theorem 3.5:
Suppose that $1/\hat{\sigma}_T \in W^{3,\infty}[0,1]$ *and* $\hat{\sigma}_T \in L^\infty[0,1]$. *Then under the definitions of* $\hat{\mathcal{T}}$ *and* $\mathcal{P}$ *above, with zero boundary conditions (*(3.3) *with* $\hat{f}_L$ *and* $\hat{f}_R$ *both zero) and using the dimensionless variables defined in Section 3.2, for any* $f \in H^4[0,1]$ *it holds that*

$$\frac{\hat{\sigma}_T(\hat{x})}{\epsilon^2}\left(\mathcal{I} - \mathcal{P}\hat{\mathcal{T}}^{-1}\hat{c}(\hat{x})\right)\hat{f}(\hat{x}) = \left(\hat{\mathcal{D}} + \mathcal{O}(\epsilon)\right)\hat{f}(\hat{x}), \tag{3.116}$$

where

$$\hat{\mathcal{D}}(\cdot) \equiv -\frac{d}{d\hat{x}}\left(\frac{1}{3\hat{\sigma}_T(\hat{x})}\frac{d}{d\hat{x}}(\cdot)\right) + \hat{\sigma}_A(\hat{x})(\cdot), \tag{3.117}$$

is an operator form of the dimensionless diffusion equation, (3.72)*, including zero boundary conditions, (*(3.73) *and* (3.74) *with* $\hat{f}_L$ *and* $\hat{f}_R$ *both zero).*

Proof.
We know

$$\hat{\mathcal{T}}^{-1} = \left(\mathcal{I} + \frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^{-1}.$$

Using (3.115) we get

$$\begin{aligned}\hat{\mathcal{T}}^{-1} &= \mathcal{I} - \frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}} + \left(\frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^2 - \left(\frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^3 + \mu^4\hat{\mathcal{T}}^{-1}\left(\frac{\epsilon}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4, \\ &= \mathcal{I} - \frac{\epsilon\mu}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}} + \epsilon^2\mu^2\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^2 - \epsilon^3\mu^3\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^3 + \mu^4\hat{\mathcal{T}}^{-1}\left(\frac{\epsilon}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4.\end{aligned}$$

Now applying $\mathcal{P}$, and noting that the operator $\hat{\mathcal{T}}^{-1}$ is only applied to functions of $\hat{x}$, we find

$$\mathcal{P}\hat{\mathcal{T}}^{-1} = \mathcal{I} + \frac{\epsilon^2}{3}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^2 + \epsilon^4\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4, \tag{3.118}$$

in which we have made use of

$$\mathcal{P}\mu^i = \begin{cases} 0\ , & \text{for } i \text{ odd}, \\ \frac{1}{i+1}, & \text{for } i \text{ even}. \end{cases}$$

Next, from our asymptotics we know that $\hat{c}(\hat{x}) = 1 - \epsilon^2\hat{\sigma}_A(\hat{x})/\hat{\sigma}_T(\hat{x}) + \mathcal{O}(\epsilon^3)$, and so

$$\begin{aligned}\mathcal{P}\hat{\mathcal{T}}^{-1}\hat{c}(\hat{x}) = \mathcal{I} + \frac{\epsilon^2}{3}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^2 - \epsilon^2\frac{\hat{\sigma}_A(\hat{x})}{\hat{\sigma}_T(\hat{x})}\mathcal{I} - \frac{\epsilon^4}{3}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^2\frac{\hat{\sigma}_A(\hat{x})}{\hat{\sigma}_T(\hat{x})} + \mathcal{O}(\epsilon^3) \\ + \epsilon^4\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{c}(\hat{x}).\end{aligned}$$

Now subtracting this from the identity operator we are left with

$$\begin{aligned}\mathcal{I} - \mathcal{P}\hat{\mathcal{T}}^{-1}\hat{c}(\hat{x}) &= \frac{-\epsilon^2}{3}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^2 + \epsilon^2\frac{\hat{\sigma}_A(\hat{x})}{\hat{\sigma}_T(\hat{x})}\mathcal{I} + \mathcal{O}(\epsilon^3) + \epsilon^4\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{c}(\hat{x}), \\ &= \frac{\epsilon^2}{\hat{\sigma}_T(\hat{x})}\hat{\mathcal{D}} + \mathcal{O}(\epsilon^3) + \epsilon^4\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{c}(\hat{x}),\end{aligned}$$

and so scaling results in

$$\frac{\hat{\sigma}_T(\hat{x})}{\epsilon^2}\left(\mathcal{I}-\mathcal{P}\hat{\mathcal{T}}^{-1}\hat{c}(\hat{x})\right)=\hat{\mathcal{D}}+\mathcal{O}(\epsilon)+\hat{\sigma}_T(\hat{x})\epsilon^2\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{c}(\hat{x}). \quad (3.119)$$

At this point we can see that the proof is almost complete, and it remains to show that the last term in (3.119) is $\mathcal{O}(\epsilon)$ if applied to a smooth enough function. This will be our sole focus for the remainder of this proof. In fact, since $\hat{c}(\hat{x})=1-\mathcal{O}(\epsilon^2)$, we just focus on just the $\mathcal{O}(1)$ multiple of the last term in (3.119) and show that

$$\epsilon^2\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{f}(\hat{x})=\mathcal{O}(\epsilon),$$

for any $\hat{f}\in H^4[0,1]$.

We start by recalling the definition of $\mathcal{T}^{-1}$, (2.28), which for zero boundary conditions says

$$\mathcal{T}^{-1}g(x)=\begin{cases}\dfrac{1}{\mu}\displaystyle\int_0^x g(x')\exp\left(\frac{-1}{\mu}\int_{x'}^x\sigma_T(z)\,\mathrm{d}z\right)\mathrm{d}x', & \mu>0,\\ -\dfrac{1}{\mu}\displaystyle\int_x^D g(x')\exp\left(\frac{1}{\mu}\int_x^{x'}\sigma_T(z)\,\mathrm{d}z\right)\mathrm{d}x', & \mu<0,\end{cases} \quad (3.120)$$

for any $g\in L^2[0,D]$. Because we are interested in the last term of (3.119), we define a function $\hat{g}$ via

$$\hat{g}(\hat{x})\equiv\hat{\sigma}_T(\hat{x})\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{f}(\hat{x}). \quad (3.121)$$

With our assumptions on $\hat{\sigma}_T$, and with $\hat{f}\in H^4$, we can verify that

$$\int_{[0,1]}\left|\hat{\sigma}_T(\hat{x})\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{f}(\hat{x})\right|^2\,\mathrm{d}\hat{x}<\infty,$$

and so $\hat{g}\in L^2[0,1]$. This allows us to say

$$\begin{aligned}\epsilon^2\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\left(\frac{1}{\hat{\sigma}_T(\hat{x})}\frac{\partial}{\partial\hat{x}}\right)^4\hat{f}(\hat{x})&=\epsilon^2\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\frac{\hat{g}(\hat{x})}{\hat{\sigma}_T(\hat{x})}\\&=\epsilon^2\mathcal{P}\mu^4\mathcal{T}^{-1}\frac{\hat{g}(\hat{x})}{\epsilon D},\end{aligned}$$

where we have used the condition (3.111) as well as the definition (3.110). Now sub-

stituting (3.120) and remembering that $x \equiv D\hat{x}$ we obtain

$$\frac{\epsilon}{D}\mathcal{P}\mu^4\mathcal{T}^{-1}\hat{g}(x/D) = \frac{\epsilon}{2D}\left[\int_0^1 \mu^3 \int_0^x \hat{g}(x'/D)\exp\left(\frac{-1}{\mu}\int_{x'}^x \sigma_T(z)\,\mathrm{d}z\right)\mathrm{d}x'\,\mathrm{d}\mu \right.$$
$$\left. - \int_{-1}^0 \mu^3 \int_x^D \hat{g}(x'/D)\exp\left(\frac{1}{\mu}\int_x^{x'} \sigma_T(z)\,\mathrm{d}z\right)\mathrm{d}x'\,\mathrm{d}\mu\right],$$

and as long as the term in square brackets is bounded above, we have the result. By making the change of variables $\mu \to -\mu$ in the second double integral, we are able to combine the two spatial integrals, leaving

$$\frac{\epsilon}{D}\mathcal{P}\mu^4\mathcal{T}^{-1}\hat{g}(x/D) = \frac{\epsilon}{2D}\int_0^D \hat{g}(x'/D)\int_0^1 \mu^3 \exp\left(\frac{-1}{\mu}\tau(x',x)\right)\mathrm{d}\mu\,\mathrm{d}x', \tag{3.122}$$

where τ is the optical path length (see Definition 2.1). Now, since we know that $-\tau(x',x)/\mu \leq 0$ for $\mu \in [0,1]$, we have $0 \leq \exp\left(-\tau(x',x)/\mu\right) \leq 1$, thus

$$\left|\epsilon^2\mathcal{P}\mu^4\hat{\mathcal{T}}^{-1}\frac{\hat{g}(\hat{x})}{\hat{\sigma}_T(\hat{x})}\right| \leq \left|\frac{\epsilon}{2D}\int_0^D \hat{g}(x'/D)\int_0^1 \mu^3\,\mathrm{d}\mu\,\mathrm{d}x'\right|$$
$$= \frac{\epsilon}{8}\left|\int_0^1 \hat{g}(\hat{x})\,\mathrm{d}\hat{x}\right| = \mathcal{O}(\epsilon), \tag{3.123}$$

which is what we were trying to show. The proof is now concluded by combining (3.123) with (3.119). □

Remark 3.6:
Theorem 3.5 was presented in the nondimensional setting introduced in Section 3.2.1. Carefully re-dimensionalising the result using the definitions from that section, we obtain the following equivalent dimensional result

$$\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right) f = \frac{1}{\sigma_T}\mathcal{D}f + \mathcal{O}(\epsilon^2), \tag{3.124}$$

in which $f(x) \equiv \hat{f}(x/D)$.

With this result we have proved that there is a relationship between the left hand side of the diffusion equation, (3.72), and a scaling of the transport equation in operator form, (3.114).

Henceforth we shall assume the cross sections are constant. Our focus for the remainder of this section will be on the right hand side of (3.114), and in Theorem 3.9

we will prove that the right hand side converges pointwise to just the source term, $\hat{Q}$, as ϵ tends to zero. To prove Theorem 3.9 we will need the following two results.

Lemma 3.7:
If

$$K(z) \equiv \frac{1}{2}\int_a^b \frac{1}{\mu} \exp\left(\frac{-|z|}{\mu}\right) \, d\mu, \tag{3.125}$$

then

$$\int_{\mathbb{R}} K(z) \, dz = b - a. \tag{3.126}$$

Proof.
The proof of this result is given in the Appendix, Section A.2. □

Lemma 3.8:
Let $f : \mathbb{R} \to \mathbb{R}$, and suppose that f is Lipschitz continuous and also globally bounded so that $\max_\xi |f(\xi)| = C$, where C is constant. Suppose also we have some function $K : \mathbb{R} \to \mathbb{R}^+$ such that

$$\int_{\mathbb{R}} K(z) \, dz = \alpha,$$

where $\alpha \in \mathbb{R}$ is some constant. Then it holds that

$$\lim_{\sigma\to\infty} \sigma \int_{\mathbb{R}} K\left(\sigma(x-y)\right) f(y) \, dy = \alpha f(x), \tag{3.127}$$

for all $x \in \mathbb{R}$.

Proof.
A proof of this result is presented in the Appendix, Section A.2. □

Using Lemmas 3.7 and 3.8 we can now prove the following result.

Theorem 3.9:
Let $Q \in L^2([x_L, x_R])$ be Lipschitz continuous, and assume that $\sigma_T = \frac{r}{\epsilon}$ where $r \in \mathbb{R}$ is constant. Then it holds that

$$\lim_{\epsilon\to 0} \sigma_T \mathcal{P}\mathcal{T}^{-1} Q(x) = Q(x), \tag{3.128}$$

pointwise, for all $x \in (x_L, x_R)$, where $\mathcal{T}$ includes zero boundary conditions (2.14) with $f_L = f_R = 0$.

Proof.
In Section 2.3 we defined the operator $\mathcal{K}_{\sigma_T}$ which describes the action $\mathcal{P}\mathcal{T}^{-1}$ for zero boundary conditions, and so we are interested in the behaviour of $\sigma_T \mathcal{K}_{\sigma_T} Q(x)$ as $\epsilon \to 0$. From (2.35) we know

$$\sigma_T \mathcal{K}_{\sigma_T} Q(x) = \sigma_T \int_{[x_L, x_R]} k_{\sigma_T}(x, y) Q(y) \, \mathrm{d}y.$$

Since our cross sections are constant, we can use (2.30) and Definition 2.1 to show that

$$k_{\sigma_T}(x, y) \equiv \frac{1}{2} \int_0^1 \frac{1}{\mu} \exp\left(\frac{-\sigma_T}{\mu} |x - y| \right) \mathrm{d}\mu.$$

Then using the function K, defined in (3.125), with $a = 0$ and $b = 1$ we see that

$$k_{\sigma_T}(x, y) = K(\sigma_T(x - y)),$$

and so

$$\sigma_T \int_{[x_L, x_R]} k_{\sigma_T}(x, y) Q(y) \, \mathrm{d}y = \sigma_T \int_{[x_L, x_R]} K\left(\sigma_T(x - y)\right) Q(y) \, \mathrm{d}y. \tag{3.129}$$

Define an extension, $\tilde{Q}$, of Q to the whole real line by

$$\tilde{Q}(x) \equiv \begin{cases} Q(x) & \text{if } x \in [x_L, x_R], \\ \left(\dfrac{x_R + \eta - x}{\eta} \right) Q(x_R) & \text{if } x \in (x_R, x_R + \eta], \\ \left(\dfrac{x - x_L + \eta}{\eta} \right) Q(x_L) & \text{if } x \in [x_L - \eta, x_L), \\ 0 & \text{else,} \end{cases} \tag{3.130}$$

for any finite $\eta > 0$. This extension is Lipschitz continuous over the whole real line and has the same global maximum as Q. It also has compact support and satisfies $\tilde{Q}\Big|_{[x_L, x_R]} \equiv Q$. Using this we now split the integral in (3.129) into two domains

$$\begin{aligned} \sigma_T \int_{[x_L, x_R]} K\left(\sigma_T(x - y)\right) Q(y) \, \mathrm{d}y = \sigma_T \int_{\mathbb{R}} K\left(\sigma_T(x - y)\right) \tilde{Q}(y) \, \mathrm{d}y - \\ \underbrace{\sigma_T \int_{\mathbb{R} \setminus [x_L, x_R]} K\left(\sigma_T(x - y)\right) \tilde{Q}(y) \, \mathrm{d}y}_{\equiv (\dagger)}. \end{aligned} \tag{3.131}$$

By Lemma 3.8 we have that

$$\sigma_T \int_{\mathbb{R}} K\left(\sigma_T(x-y)\right) \tilde{Q}(y)\, \mathrm{d}y \to \tilde{Q}(x) = Q(x), \quad \forall x \in [x_L, x_R] \text{ as } \sigma_T \to \infty, \tag{3.132}$$

where in this case we knew $\alpha = 1$ by Lemma 3.7. It remains to show that $(\dagger) \to 0$ as $\sigma_T \to \infty$. To do this we look at the two halves of the domain of integration of $(\dagger)$ in turn

$$(\dagger) = \underbrace{\sigma_T \int_{(x_R,\infty)} K\left(\sigma_T(x-y)\right) \tilde{Q}(y)\, \mathrm{d}y}_{\equiv(\dagger^+)} + \underbrace{\sigma_T \int_{(-\infty,x_L)} K\left(\sigma_T(x-y)\right) \tilde{Q}(y)\, \mathrm{d}y}_{\equiv(\dagger^-)}. \tag{3.133}$$

First of all, applying the change of variables $z = \sigma_T(x-y)$, we get

$$\begin{aligned} |(\dagger^+)| &= \left| \sigma_T \left(\frac{-1}{\sigma_T}\right) \int_{(\sigma_T(x-x_R),-\infty)} K(z) \tilde{Q}(x - {\sigma_T}^{-1} z)\, \mathrm{d}z \right| \\ &= \left| \int_{(-\infty,\sigma_T(x-x_R))} K(z) \tilde{Q}(x - {\sigma_T}^{-1} z)\, \mathrm{d}z \right|. \end{aligned} \tag{3.134}$$

Since $\tilde{Q}$ has a global bound (attained within the interval $[x_L, x_R]$) we can write

$$\left|(\dagger^+)\right| \leq \left\|\tilde{Q}\right\|_{\infty,[x_L,x_R]} \int_{(-\infty,\sigma_T(x-x_R))} K(z)\, \mathrm{d}z. \tag{3.135}$$

Provided that $x \in [x_L, x_R)$, the upper limit tends to $-\infty$ and so the integral tends to zero as $\sigma_T \to \infty$. Using an equivalent argument we can also find that $|(\dagger^-)| \to 0$ as $\sigma_T \to \infty$ provided that $x \in (x_L, x_R]$. Thus taking $x \in (x_L, x_R)$ satisfies both conditions, and together with (3.132) this concludes the proof. □

We can immediately apply this result to the right hand side of (3.114), scaled by $\hat{\sigma}_T/\epsilon^2$, to obtain the following.

Corollary 3.10:
Assume $\hat{\sigma}_T$ is constant. In Section 3.2.2 we found that $\hat{Q}(\hat{x}) = \epsilon^2 \hat{Q}_2(\hat{x}) + \mathcal{O}(\epsilon^3)$, and in (3.110) *we defined*

$$\hat{\mathcal{T}}^{-1}(\cdot) \equiv \mathcal{T}^{-1}\left(\frac{\hat{\sigma}_T}{\epsilon D}(\cdot)\right).$$

Using these, it holds that

$$\lim_{\epsilon \to 0} \frac{\hat{\sigma}_T}{\epsilon^2} \mathcal{P} \hat{\mathcal{T}}^{-1} \hat{Q}(\hat{x}) = \hat{\sigma}_T \hat{Q}_2(\hat{x}), \tag{3.136}$$

pointwise, for all $\hat{x} \in [0,1]$.

Proof.
Using the definition of $\hat{\mathcal{T}}^{-1}$ we know immediately that

$$\begin{aligned}\frac{\hat{\sigma}_T}{\epsilon^2}\mathcal{P}\hat{\mathcal{T}}^{-1}\hat{Q}(\hat{x}) &= \frac{\hat{\sigma}_T}{\epsilon^2}\mathcal{P}\mathcal{T}^{-1}\frac{\hat{\sigma}_T}{D\epsilon}\hat{Q}(\hat{x}) \\ &= \frac{\hat{\sigma}_T}{D\epsilon}\mathcal{P}\mathcal{T}^{-1}\frac{\hat{\sigma}_T}{\epsilon^2}\hat{Q}(\hat{x}).\end{aligned}$$

Substituting in the source term expansion, $\hat{Q}(\hat{x}) = \epsilon^2\hat{Q}_2(\hat{x}) + \mathcal{O}(\epsilon^3)$, we are left with

$$\frac{\hat{\sigma}_T}{D\epsilon}\mathcal{P}\mathcal{T}^{-1}\frac{\hat{\sigma}_T}{\epsilon^2}\hat{Q}(\hat{x}) = \frac{\hat{\sigma}_T}{D\epsilon}\mathcal{P}\mathcal{T}^{-1}\hat{\sigma}_T\left(\hat{Q}_2(\hat{x}) + \mathcal{O}(\epsilon)\right).$$

Therefore, applying Theorem 3.9 (where in this case $r = \hat{\sigma}_T/D$) we obtain the desired result. □

To understand what the two new results (Theorem 3.5 and Theorem 3.9) can tell us about the DSA algorithm we look again at a block operator equation. This time we will work from the source iteration algorithm (Algorithm 1), which we summarise here as

$$\begin{aligned}\mathcal{T}\psi^{(k+1/2)} &= \sigma_S\phi^{(k)} + Q, \\ \phi^{(k+1/2)} &= \mathcal{P}\psi^{(k+1/2)},\end{aligned}$$

subject to boundary conditions, (2.14). Writing this in a block operator matrix form results in

$$\begin{pmatrix} \mathcal{T} & 0 \\ -\mathcal{P} & \mathcal{I} \end{pmatrix}\begin{pmatrix} \psi^{(k+1/2)} \\ \phi^{(k+1/2)} \end{pmatrix} = \begin{pmatrix} 0 & \sigma_S\mathcal{I} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \psi^{(k)} \\ \phi^{(k)} \end{pmatrix} + \begin{pmatrix} Q \\ 0 \end{pmatrix}. \tag{3.137}$$

We start by subtracting this from the exact expression in order to obtain an error form,

$$\begin{pmatrix} \mathcal{T} & 0 \\ -\mathcal{P} & \mathcal{I} \end{pmatrix}\begin{pmatrix} e^{(k+1/2)} \\ E^{(k+1/2)} \end{pmatrix} = \begin{pmatrix} 0 & \sigma_S\mathcal{I} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} e^{(k)} \\ E^{(k)} \end{pmatrix}, \tag{3.138}$$

where

$$\begin{aligned}e^{(k)} &\equiv \psi - \psi^{(k)}, \\ E^{(k)} &\equiv \phi - \phi^{(k)}.\end{aligned}$$

Subtracting $\left(\begin{smallmatrix} 0 & \sigma_S\mathcal{I} \\ 0 & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} e^{k+1/2} \\ E^{k+1/2} \end{smallmatrix}\right)$ from both sides we have

$$\left[\begin{pmatrix} \mathcal{T} & 0 \\ -\mathcal{P} & \mathcal{I} \end{pmatrix} - \begin{pmatrix} 0 & \sigma_S\mathcal{I} \\ 0 & 0 \end{pmatrix}\right]\begin{pmatrix} e^{(k+1/2)} \\ E^{(k+1/2)} \end{pmatrix} = \begin{pmatrix} 0 & \sigma_S\mathcal{I} \\ 0 & 0 \end{pmatrix}\left[\begin{pmatrix} e^{(k)} \\ E^{(k)} \end{pmatrix} - \begin{pmatrix} e^{(k+1/2)} \\ E^{(k+1/2)} \end{pmatrix}\right],$$

$$\Rightarrow \begin{pmatrix} \mathcal{T} & -\sigma_S\mathcal{I} \\ -\mathcal{P} & \mathcal{I} \end{pmatrix}\begin{pmatrix} e^{(k+1/2)} \\ E^{(k+1/2)} \end{pmatrix} = \begin{pmatrix} 0 & \sigma_S\mathcal{I} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} e^{(k)} - e^{(k+1/2)} \\ E^{(k)} - E^{(k+1/2)} \end{pmatrix}.$$

Lastly, we apply the same Gaussian elimination process as in (3.113) via pre-multiplying by $\left(\begin{smallmatrix} \mathcal{I} & 0 \\ \mathcal{P}\mathcal{T}^{-1} & \mathcal{I} \end{smallmatrix}\right)$ to obtain

$$\begin{pmatrix} \mathcal{T} & -\sigma_S\mathcal{I} \\ 0 & \mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S \end{pmatrix}\begin{pmatrix} e^{(k+1/2)} \\ E^{(k+1/2)} \end{pmatrix} = \begin{pmatrix} 0 & \sigma_S\mathcal{I} \\ 0 & \mathcal{P}\mathcal{T}^{-1}\sigma_S \end{pmatrix}\begin{pmatrix} e^{(k)} - e^{(k+1/2)} \\ E^{(k)} - E^{(k+1/2)} \end{pmatrix}. \tag{3.139}$$

The bottom line of (3.139) is precisely the earlier equation (3.78) (where $\mathcal{P}\mathcal{T}^{-1}\sigma_S \equiv \mathcal{K}_{\sigma_T}\sigma_S$ for zero boundary conditions) from our description of a general synthetic acceleration method for solving the transport equation. Namely we have

$$\begin{aligned}\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S\right)E^{(k+1/2)} &= \mathcal{P}\mathcal{T}^{-1}\sigma_S\left(E^{(k)} - E^{(k+1/2)}\right) \\ &= \mathcal{P}\mathcal{T}^{-1}\sigma_S\left(\phi^{(k+1/2)} - \phi^{(k)}\right).\end{aligned} \tag{3.140}$$

Applying our new Theorem 3.5 to the left hand side of (3.140) and multiplying through by σ_T we have that

$$\mathcal{D}E^{(k+1/2)} + \mathcal{O}(\epsilon) = \sigma_T\mathcal{P}\mathcal{T}^{-1}\sigma_S\left(\phi^{(k+1/2)} - \phi^{(k)}\right).$$

We then know by Theorem 3.9 that for small enough ϵ (and for constant cross sections) the error, $E^{(k+1/2)}$, can be approximated by

$$\mathcal{D}\tilde{E}^{(k+1/2)} = \sigma_S\left(\phi^{(k+1/2)} - \phi^{(k)}\right), \tag{3.141}$$

which is the approximate additive correction used in DSA (Algorithm 3).

To conclude this section we recall Lemma 3.3 where we saw that DSA was equivalent to a preconditioned Richardson scheme. It was noted that this scheme would converge quickly if $(\mathcal{I}+\mathcal{D}^{-1}\sigma_S(x))\left(\mathcal{I} - \mathcal{P}\mathcal{T}^{-1}\sigma_S(x)\right) \approx \mathcal{I}$, and in light of our work in this section we can now comment on this point. Theorem 3.5 showed (in a dimensionless setting)

that

$$\frac{\hat{\sigma}_T(\hat{x})}{\epsilon^2}\left(\mathcal{I}-\mathcal{P}\hat{\mathcal{T}}^{-1}\hat{c}(\hat{x})\right)\hat{f}(\hat{x})=\left(\hat{\mathcal{D}}+\mathcal{O}(\epsilon)\right)\hat{f}(\hat{x}),$$

for a suitably smooth function, $\hat{f}$. Using this, some manipulation with careful reference to the nondimensional definitions of Section 3.2.1 results in, for constant cross sections,

$$\left(\mathcal{I}+\mathcal{D}^{-1}\sigma_S\right)\left(\mathcal{I}-\mathcal{P}\mathcal{T}^{-1}\sigma_S\right)=\mathcal{I}+\mathcal{O}(\epsilon^2)+\mathcal{O}(\epsilon^3)(\mathcal{I}+\mathcal{D}^{-1}\sigma_S). \tag{3.142}$$

We also know that $\sigma_S=\mathcal{O}(\epsilon^{-1})$ and $\mathcal{D}=\mathcal{O}(\epsilon)$. This indicates that the last term in (3.142) behaves as $\mathcal{O}(\epsilon)$, and so

$$\left(\mathcal{I}+\mathcal{D}^{-1}\sigma_S\right)\left(\mathcal{I}-\mathcal{P}\mathcal{T}^{-1}\sigma_S\right)=\mathcal{I}+\mathcal{O}(\epsilon).$$

Even though this argument is heuristic, it suggests we should see DSA perform better as ϵ decreases. This agrees with the theory in Section 3.2 and will be demonstrated numerically in Section 3.5.

Summary

In this section we started with a nondimensional Schur complement equation for the scalar flux, (3.114). We then proved two results, Theorem 3.5 and Theorem 3.9, which concerned the left and right sides of this Schur complement form respectively. When used together these two results show that as $\epsilon\to 0$, the Schur complement form converges to the diffusion equation, (3.72), derived in Section 3.2. Furthermore, working from an operator matrix form of source iteration we can replicate the derivation of a synthetic acceleration method that was illustrated in Section 3.3.1. Applying our new theorems to the resulting block operator form yields the basic form of DSA (Algorithm 3). However, currently this is only proved for constant cross sections and under strict smoothness assumptions on the scalar flux.

Nonetheless, the results in this section present a new way of linking the diffusion approximation to the scalar flux, (3.72), and the Schur complement form of the neutron transport equation, (3.114). Future improvements to this work could result in a derivation of the diffusion equation independent of the asymptotic approach. Such a derivation would have the potential to provide further insight into the use of DSA as a preconditioner if the operator preconditioner of [27] and [7] could be obtained from it. However for now the restrictive nature of the required assumptions and reliance upon the earlier asymptotic work mean it cannot replace other derivations of the diffusion equation.

We concluded by giving a heuristic argument for why DSA should perform better as ϵ decreases. As mentioned this will be demonstrated numerically in Section 3.5, however we will also see that DSA converges faster than source iteration over the whole tested range of epsilon (see Section 3.5.1). Intuitively, even if diffusion isn't the dominant process of neutron transport in the domain, there will still be some small amount of diffusive behaviour and so a diffusion approximation will always add some improvement to the current approximate scalar flux. Our block analysis might provide an avenue for a mathematical justification of this physically intuitive statement. Whether the improvement is big enough to justify the added cost of DSA over basic source iteration (or some other method) requires a cost analysis of the kind presented in Adams and Larsen [2, Section II.B] and is not attempted here.

There is certainly plenty of scope for further exploration of diffusion synthetic acceleration from a block operator standpoint. In particular it is likely that the results presented in this section hold in higher dimensions also, and with less restrictive assumptions.

3.5 Numerical Results

We will conclude this chapter by carrying out some 2D numerical tests aimed at observing the advantages of diffusion synthetic acceleration over source iteration. To do this we will be numerically solving the neutron transport equation, with zero incoming boundary conditions, over a simple 2D spatial domain and a 1D angular domain. We will also be solving a two dimensional version of the diffusion equation derived in Section 3.2. This is given by

$$-\nabla \cdot \left(\frac{1}{3\sigma_T(\mathbf{r})} \nabla \phi(\mathbf{r}) \right) + \sigma_A(\mathbf{r})\phi(\mathbf{r}) = Q(\mathbf{r}),$$

with $\mathbf{r} \in V \subset \mathbb{R}^2$, subject to the 2D Robin boundary condition

$$\frac{\lambda}{\sigma_T(\mathbf{r})} \hat{\mathbf{n}} \cdot \nabla \phi(\mathbf{r}) + \phi(\mathbf{r}) = 0, \quad \forall \mathbf{r} \in \partial V,$$

where $\lambda \approx 0.710446$ (see [40, Section 4]) and where $\hat{\mathbf{n}}$ is the outward pointing unit normal to the boundary at $\mathbf{r}$. This boundary condition is of the general form given by Bell and Glasstone [12, Section 3.1e], however we have used the coefficient λ/σ_T taken from the 1D boundary conditions found in Section 3.2.5. To solve the 2D diffusion equation numerically we will use continuous finite elements in space over the same uniform mesh and spatial resolution as for the transport equation. In Chapter 5 we cover the details of a discontinuous Galerkin finite element discretisation of the transport equation (and

we will assume some knowledge of that chapter during this section) however we do not cover the continuous finite element discretisation used for the diffusion equation. For information on this, see for example Brenner and Scott [16].

We will start in Section 3.5.1 by directly comparing the convergence of SI and DSA for a range of values of the scattering ratio. We will use two different methods to vary the scattering ratio: firstly by varying the absorption cross section (as in Section 2.7) while keeping other parameters fixed; secondly by using the asymptotic variable, ϵ, to parametrise the cross sections and source, and vary ϵ towards zero. Though there is slightly different behaviour in each case, generally we will observe that for scattering ratios less than one DSA never degrades and often improves the rate of convergence of SI. However if we allow the scattering ratio to increase above one then DSA can converge more slowly than SI (see Figure 3-1). After that we will focus on the convergence of the two algorithms with respect to the parameter ϵ. We will consider the limit $\epsilon \to 0$ and will see that DSA can converge quickly in situations where SI converges prohibitively slowly. However in Section 3.5.2 we will observe that the convergence of DSA is not straightforward.

3.5.1 Comparing SI and DSA

In our first two numerical experiments we will compare the convergence rates of source iteration and diffusion synthetic acceleration (DSA) for different values of the scattering ratio. We will use a square domain, $V = [0,1] \times [0,1]$, spatial resolutions $M_x = M_y = 8$ and angular resolution $N = 23$. For each value of the scattering ratio we will run both SI and DSA to a tolerance of 10^{-12} or for 15 iterations (whichever occurs sooner) and will measure the ratio between the errors in the last two iterates for each method. By comparing these error ratios we will be able to see which method is converging faster and understand how that rate of convergence is related to the scattering ratio.

As mentioned above, we will use two different methods to vary the scattering ratio. Our first method will be to vary the absorption cross section (as in Section 2.7) whilst fixing the total cross section, $\sigma_T = 1$, and source, $Q = 1$. We will vary the absorption cross section, σ_A, from 0.9 down to -3.4, causing the scattering ratio to vary from 0.1 up to 4.4. As discussed in Section 2.7.1, values of the scattering ratio over 1 can be interpreted as a very basic inclusion of fission into the system.

In Table 3.1 for each value of the scattering ratio we have given the observed error ratio after a fixed number of iterations (we used 15) for source iteration (column 2) and for DSA (column 3). These values are also plotted in Figure 3-1.

We focus first on Table 3.1 for values of the scattering ratio less than 1. In this range, both methods converged and DSA consistently had the faster rate of convergence, even

Scattering Ratio, σ_S/σ_T	Observed error ratio on convergence SI	DSA
0.10	0.036	0.008
0.20	0.072	0.016
0.30	0.108	0.024
0.40	0.144	0.033
0.50	0.180	0.041
0.60	0.216	0.048
0.70	0.252	0.054
0.80	0.288	0.067
0.90	0.324	0.075
1.00	0.360	0.074
1.50	0.541	0.162
2.00	0.721	0.415
2.50	0.901	18.763
3.00	1.081	0.490
3.50	1.261	0.405
4.00	1.441	0.354

Table 3.1: Table of the observed error ratios of source iteration and DSA for varying scattering ratio.

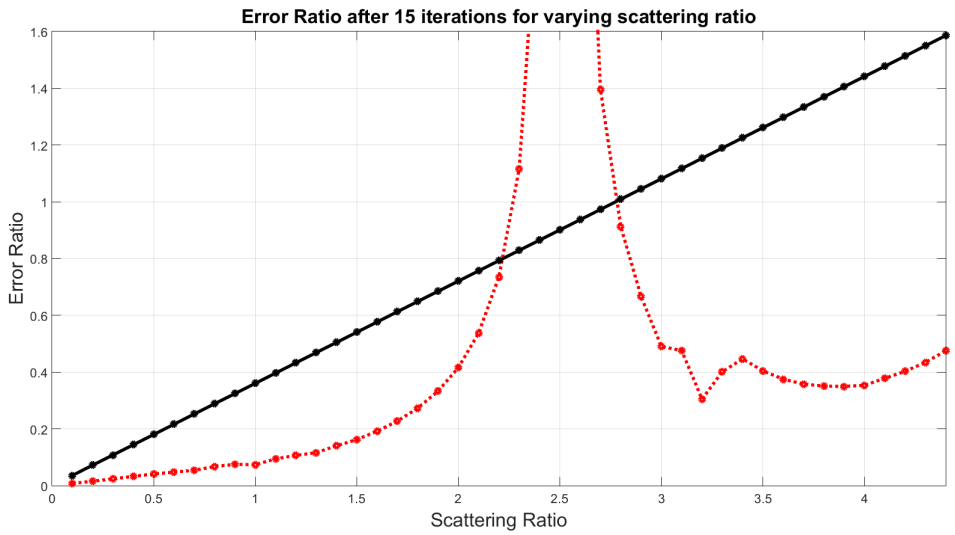

Figure 3-1: Plot of the observed error ratio versus scattering ratio for source iteration (black, solid line) and DSA (red, dotted line).

when the scattering ratio was small. The biggest difference in convergence rate was when the scattering ratio was close to 1, which is exactly the situation that DSA is targeted towards. Though the example was very basic, this does lend support to the use of DSA in overcoming source iteration's deficiencies in scattering dominated domains. Furthermore it shows that DSA can improve the convergence of source iteration for all values of the scattering ratio, and if we only cared about the number of iterations needed to converge, DSA would always be worth implementing. However the added computational cost of implementing DSA means that in reality SI can be the faster method for many problems (see Adams and Larsen [2, Section II.B]). A more physically realistic example motivated by a real-world problem will be considered at the end of Chapter 4 which will reinforce these conclusions.

Focussing now on the range of scattering ratios greater than 1, we observe that for ratios between 2.3 and 2.7 DSA had a higher error ratio than source iteration. To understand this we recall Remark 3.2 where we noted that the strictly positive absorption cross section implied the diffusion operator was positive-definite. In this test we have allowed σ_A to drop below zero and so *shift* the eigenvalues of the diffusion operator downwards. For values of this shift close to eigenvalues of $\mathcal{D}$, the operator becomes nearly singular leading to the observed divergence. The same behaviour was present for other mesh resolutions, and so this effect is not caused by discretisation. In allowing the scattering ratio to exceed 1 we were very basically including the fission interaction. However whilst this is somewhat appropriate for source iteration, including fission interactions in DSA is more complicated. For a version of DSA including fission (called *fission DSA* or FDSA) see T. J. Urbatsch [71, Chapter VII].

We note lastly that for scattering ratios over 2.7 DSA converges once more, despite the divergence of source iteration. It is hard to draw physically meaningful insight from this since the considered range of scattering ratios arises from non-physical material data. Furthermore it seems likely that once the absorption cross section decreases further it will coincide with another eigenvalue of the diffusion operator, leading to singularity and divergence once more. Nonetheless it demonstrates the impact that the diffusion step of DSA has on the convergence of source iteration. Potentially, in this situation a method solely using the diffusion approximation would be more accurate than either DSA or source iteration.

In our second test we will vary the scattering ratio by parametrising the cross sections and source in terms of the asymptotic parameter, ϵ, and so is consistent with the asymptotic theory in Section 3.2. Specifically we will define

$$\sigma_T = 1/\epsilon, \qquad \sigma_A = \epsilon, \qquad Q = \epsilon,$$

Epsilon value	Corresponding scattering ratio	Observed error ratio on convergence	
		SI	DSA
0.97	0.05	0.018	0.004
0.92	0.15	0.057	0.012
0.87	0.25	0.100	0.020
0.81	0.35	0.147	0.030
0.74	0.45	0.200	0.038
0.67	0.55	0.261	0.044
0.59	0.65	0.333	0.057
0.50	0.75	0.423	0.075
0.39	0.85	0.547	0.109
0.22	0.95	0.756	0.191

Table 3.2: Table of the observed error ratios of source iteration and DSA for varying epsilon.

and therefore the scattering ratio is given by

$$\frac{\sigma_S}{\sigma_T} = \frac{\frac{1}{\epsilon} - \epsilon}{\frac{1}{\epsilon}} = 1 - \epsilon^2. \tag{3.143}$$

We choose a range of 19 values of ϵ such that the corresponding values of the scattering ratio range evenly from 0.05 up to 0.95. By choosing the material data in this way we are directly considering the transition between a transport dominated regime and a diffusion dominated regime. Since the diffusion equation best approximates the scalar flux for small ϵ, we should expect to see DSA greatly outperform SI as the scattering ratio (3.143) approaches 1.

The data from this experiment are given in Table 3.2, where as before for each value of the scattering ratio we give the observed error ratio after 15 iterations for source iteration (column 2) and for DSA (column 3). We also give this data graphically in Figure 3-2, where the horizontal axis shows the corresponding scattering ratio and the vertical axis shows the observed error ratio.

We can immediately observe that, as in our first test, DSA consistently converged faster than source iteration over the tested range. This is the outcome we expected since we deliberately used material properties that align with the requirements for the diffusion equation to provide a good approximation to the scalar flux (see Section 3.2). However in contrast to our first test, the observed error ratio of source iteration does not depend linearly on the scattering ratio. Obtaining such different behaviour between these two tests demonstrates that the scattering ratio is not the only indicator of the convergence of source iteration. For constant material data this is reflected in our

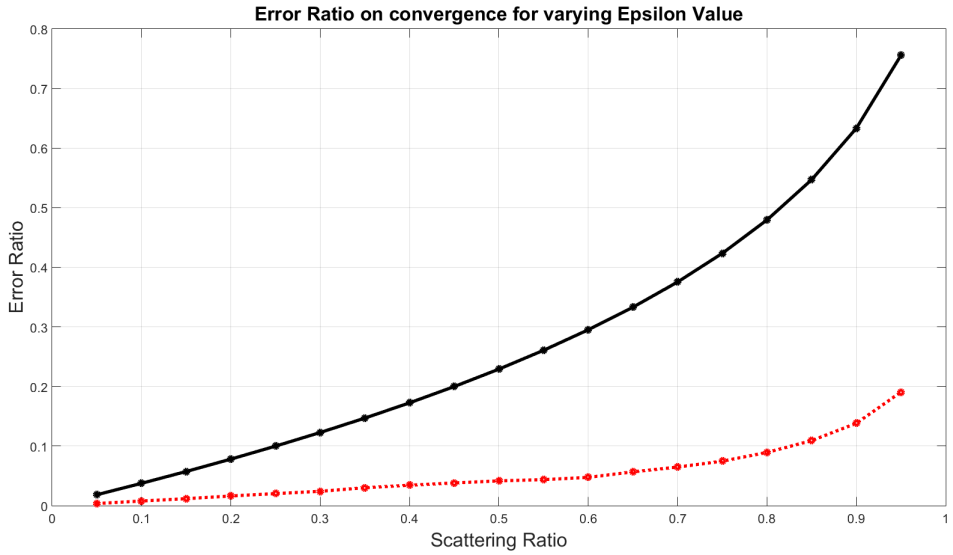

Figure 3-2: Plot of the observed error ratio versus scattering ratio (induced by varying epsilon) for source iteration (black, solid line) and DSA (red, dotted line).

source iteration convergence result, Corollary 2.26, where the domain width and size of the total cross section come into play.

The convergence behaviour of source iteration suggests that if we were to consider the limit $\epsilon \to 0$, the observed error ratio would increase towards 1. Indeed this is the case and in the above example, for a value of $\epsilon = 0.01$ source iteration would already need almost 6000 iterations to converge to a tolerance of just 10^{-4}. It is this limiting behaviour that we will explore for DSA in the final test of this chapter.

We end again by noting that the same convergence behaviour was present for other tested mesh resolutions, and so is not caused by the discretisation.

3.5.2 DSA in the Epsilon Limit

In this final experiment we will consider only diffusion synthetic acceleration and (as in our second experiment above) we will parametrise the total cross section, absorption cross section and the source in terms of the asymptotic variable, ϵ. However for this experiment we will allow $\epsilon \to 0$ exponentially in order to more fully explore the behaviour of DSA in this limit.

As before, we specify $\sigma_T = 1/\epsilon$, $\sigma_A = \epsilon$ and $Q = \epsilon$, over the unit square domain $V = [0,1] \times [0,1]$. We will use the spatial resolutions $M_x = M_y = 16$ and angular resolution $N = 32$. For each value of ϵ we will run DSA until it converges to a tolerance of 10^{-5}. For this experiment we will record the ratio between consecutive

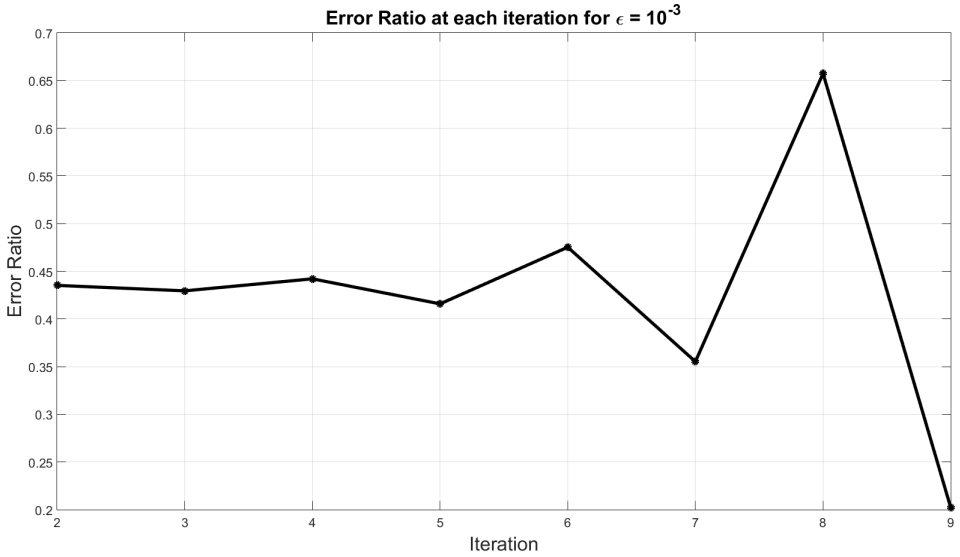

Figure 3-3: An example of how the observed error ratio may oscillate during the DSA iterations. This plot is for $\epsilon = 10^{-3}$.

errors for the last *two* iterations before convergence, or after a maximum of 25 iterations. This is because for certain values of ϵ we will see oscillation in the error ratio as the iterations proceed and so considering just one ratio is misleading. For an example of this oscillation, see Figure 3-3 which plots the error ratio at each iteration for $\epsilon = 10^{-3}$. The last two data points from this graph are listed in Table 3.3 (line 6).

In Table 3.3, for a subset of the tested epsilon values (column 1) we have given the observed error ratio of DSA upon convergence (column 2) and one step prior to convergence (column 3). In Figure 3-4 we have plotted the same data over the full range of tested epsilon values.

Looking at the data we first notice that DSA converged over the whole tested range of epsilon. In particular the error ratio remained below 0.7 despite the oscillatory behaviour that can be seen in Figure 3-4. Using source iteration to solve over this range of epsilon is infeasibly slow, and so these results support the use of DSA to overcome the deficiencies exhibited by source iteration.

Next, we can see that for epsilon values between about 10^{-2} and 10^{-5} the convergence of DSA is oscillatory. The amplitude of the oscillation can be reduced by refining the spatial mesh, which we observed by running successive calculations with $M_x = M_y = 2$, 4, 8 and 16. Therefore it is possible that by taking a sufficiently fine mesh the oscillation may be eliminated, however due to hardware restrictions it was not possible to refine further than $M_x = M_y = 16$ for this test. Indeed recalling the

Epsilon	Observed error ratio of DSA, measured: at convergence	one iteration before convergence
$10^{-0.5}$	0.141	0.141
$10^{-1.0}$	0.314	0.314
$10^{-1.5}$	0.399	0.398
$10^{-2.0}$	0.429	0.415
$10^{-2.5}$	0.261	0.581
$10^{-3.0}$	0.202	0.657
$10^{-3.5}$	0.264	0.541
$10^{-4.0}$	0.363	0.474
$10^{-4.5}$	0.449	0.432
$10^{-5.0}$	0.441	0.435

Table 3.3: Table of the observed error ratio of DSA, measured at convergence and one iteration before convergence, for varying epsilon values.

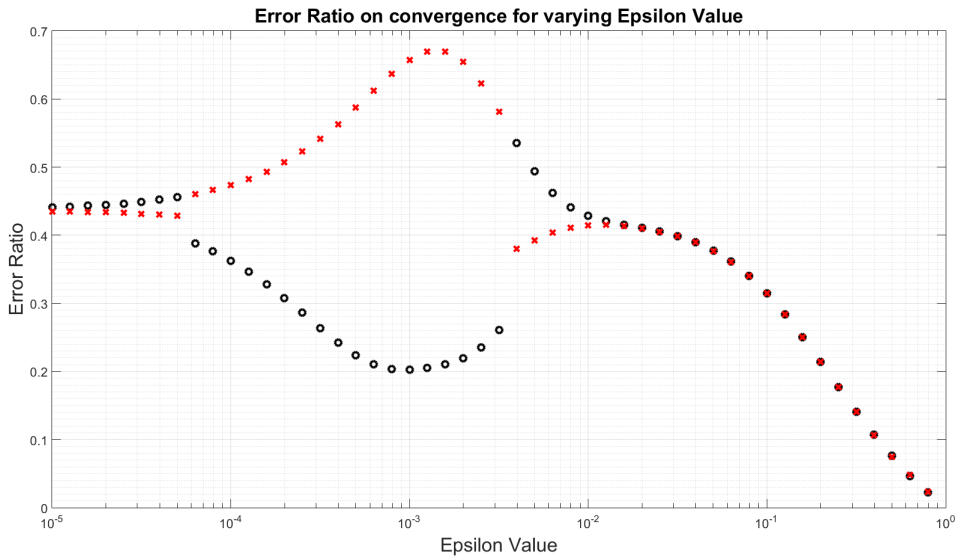

Figure 3-4: Plot of the observed error ratio of DSA, measured at convergence (black circles) and one iteration before convergence (red crosses), versus epsilon.

asymptotic theory from Section 3.2, a mesh fine enough to capture the boundary layer behaviour, or one refined locally near the boundaries, may be necessary.

Alternatively the cause of these oscillations might be our choice of discretisation. The convergence of DSA is not straightforward and has been studied extensively since its original development. In particular the discretisation of the diffusion equation is important (see [60], [4] and [5]) and must be so-called *consistent* with that of the transport equation. This idea was originally presented by Alcouffe [4, 5], where it is explained that the discretisations of the transport and diffusion equations cannot be independent, and that instead the discrete diffusion equation must be derived from a discretised form of the transport equation. Larsen [51] gives a summary of the concept of *consistency* as well as developments that led up to Alcouffe's work, and he goes on to present a so-called *four-step method* that allows him to derive unconditionally stable acceleration methods that are appropriate for several different spatial differencing methods. However the four-step method does not extend to advanced schemes such as discontinuous finite element methods (DFEM), which led to the development of *partially consistent* methods such as the *modified four-step* (M4S) DSA method of Adams and Martin [3]. Ultimately fully consistent methods for DFEM were developed (see for example Warsa et. al. [73]) however the partially consistent methods were seen to perform better in some situations.

Partially consistent methods give up the some of the stability or effectiveness of fully consistent methods, however they gain other attributes that make them desirable. For example, M4S DSA yields a smaller linear system than the fully consistent equivalent, leading to cheaper iterations. Alternatively Wareing et. al. [72] obtain a partially consistent scheme that uses a symmetric positive-definite, continuous discretisation of the diffusion equation, so the resulting DSA method can utilise the conjugate gradients method and thus be very efficient.

We are using discontinuous Galerkin finite elements to solve the transport equation and continuous finite elements to solve the diffusion equation. Whilst we are using the same mesh and the same linear basis elements, it is possible that these oscillations are the result of only partial consistency in our method.

Conclusion

To conclude, through these numerical results we have seen the following:

- source iteration performs well for scattering ratios close to zero (as seen in Section 2.7);
- for scattering ratios close to one, DSA performs well;

- DSA always improves upon the rate of convergence of source iteration for physically valid material data, even when the scattering ratio is close to zero.

Since DSA is always more computationally expensive (per iteration) than source iteration, it might not be desirable to employ the extra power of DSA when a computationally cheaper method works well enough (see Adams and Larsen [2, Section II.B]). As we will see in Chapter 4, it can also be the case that only a small part of the domain contains material whose properties suggest a need for DSA to be used. In such a situation, applying the more costly DSA method over the whole domain just to resolve an issue in a small part seems unnecessary, though not doing so can lead to misleading solutions with localised unresolved errors.

Our work in Chapter 4 is motivated by these situations. We develop two domain decomposition implementations of source iteration, and later (in Section 4.5) see how this new framework allows us to apply DSA locally only where it is most needed. We will see that this approach can indeed resolve the problem explained above, as well as opening the door to the local application of other numerical methods.

Chapter 4

Domain Decomposition

Contents

4.1 Background and Motivation 111
4.2 Domain Decomposed Source Iteration (DDSI) 113
4.2.1 Source Iteration Recap 114
4.2.2 Jacobi DDSI 115
4.2.3 Gauss-Seidel DDSI 119
4.3 Convergence of Gauss-Seidel DDSI 124
4.4 Convergence of Jacobi DDSI 130
4.4.1 Bounding the Solution Operator Norm 131
4.4.2 Convergence of Jacobi DDSI 138
4.5 Numerical Tests 145
4.5.1 Verifying Jacobi DDSI Convergence 146
4.5.2 Verifying Gauss-Seidel DDSI Convergence 148
4.5.3 Physically Motivated Example: Spent Fuel Pool 153

4.1 Background and Motivation

In this thesis we have looked at the convergence of different iterative methods when they are used to solve the neutron transport equation, as given in Section 2.2. In particular, in Chapter 2 we saw how the convergence of source iteration is heavily tied to the maximum scattering ratio of the domain. Consequently for domains with a small scattering ratio source iteration converges quickly, but if the domain contains so-called diffusive regions (characterised by a high scattering ratio) then other faster methods are required.

One such faster method was considered in Chapter 3, and uses a specific diffusion equation to improve upon the output of source iteration at each step. This diffusion equation approximates the scalar flux and is most accurate in a diffusive domain. In this way it can counterbalance the slow convergence of source iteration, and combining both leads to the method known as diffusion synthetic accelerated source iteration (DSA) given in Section 3.3.2. DSA converges quickly over all values of the scattering ratio, as was demonstrated in Chapter 3.

This improvement does not come for free and one down side to DSA is that it is more computationally expensive to implement than source iteration: each iteration requires the source iteration step plus solving a diffusion equation. Therefore it would be useful to limit its application only to the parts of the domain in which it will have the greatest effect. This is the driving motivation behind the work in this chapter, in which we will develop domain decomposed iterative methods for solving the transport equation.

Domain decomposition (DD) methods are used to solve partial differential equations by decomposing the domain into several subdomains. The methods involve solving subproblems on these subdomains while enforcing suitable continuity requirements between the subdomains. DD methods lend themselves to parallelisation, and have advantages in handling complex or irregular geometries. There are many different direct DD solvers, however we will focus on developing iterative DD algorithms for solving the transport equation. Iterative DD methods were originally proposed by H. A. Schwarz in 1870 [68] however interest in these methods picked up during the 1980s as parallel computing architectures became more common, and has continued growing since. Iterative DD methods can be broadly broken into two types: *overlapping* methods, in which the intersection of adjacent subdomains is non-empty, and *non-overlapping* methods, in which all the subdomains are disjoint. The methods we will propose fall into the latter category, meaning the subdomains are only able to communicate across their shared boundaries. The precise details of how this communication takes place is the source of the main difference between the two methods we will describe. Further information on domain decomposition methods can be found in Chan and Mathew [21].

As mentioned in Section 1.3 domain decomposition methods have been applied to solving the neutron transport equation before, however the focus has been mostly towards parallelisation. Yavuz and Larsen [76] [77] gave two algorithms similar to those that we will develop in Section 4.2, however there are differences that we will identify after each method has been presented (see Algorithms 6 and 8). Furthermore in Sections 4.3 and 4.4 we prove new results (namely Theorems 4.4 and 4.8) showing that the two domain decomposition algorithms converge, and we know of no other work

that achieves this.

The chapter will be structured as follows. Firstly we will start in Section 4.2.1 by re-introducing the transport equation and recapping the basic source iteration algorithm. From there in Sections 4.2.2 and 4.2.3 we will build two different domain decomposition source iteration (DDSI) algorithms: namely *Jacobi DDSI* and *Gauss-Seidel DDSI*. The former will be seen to be more suited to parallelisation, but suffers a slightly worse rate of convergence. The latter is less amenable to parallelisation but under some assumptions is equivalent to the full source iteration algorithm. Sections 4.3 and 4.4 will state and prove convergence results for both algorithms under the assumption of convex subdomains. Lastly in Section 4.5 we will carry out some numerical experiments to confirm our results and to illustrate further properties of the different methods. We will also demonstrate the advantages of applying DSA only to specific parts of the domain through the use of a physically motivated 'fuel pool' example.

4.2 Domain Decomposed Source Iteration (DDSI)

In this section we will ultimately introduce and explain two domain decomposition source iteration (DDSI) algorithms: Jacobi DDSI and Gauss-Seidel DDSI. These both solve the transport equation over a domain divided into an arbitrary number of subdomains. On each of the subdomains we will have an approximation to the scalar flux, ϕ; this is our *current iterate* (or iteration k). We want to define methods of using the current iterate to obtain a better approximation to ϕ on each subdomain (called iteration $k+1$).

The two algorithms we will define differ in the way they pass information across boundaries shared by neighbouring subdomains, and they have different advantages and disadvantages resulting from this. We will prove that both of these methods converge, though will highlight limitations of each convergence proof in turn. In particular we will show that by paying close attention to how the angular variable affects the flow of information through the domain, the Gauss-Seidel DDSI method is equivalent to source iteration applied over the whole domain. Consequently, under the assumptions of Theorem 4.4, Gauss-Seidel DDSI inherits the convergence properties of full SI, which were proved in Chapter 2. The primary disadvantage of Gauss-Seidel versus Jacobi DDSI is that it is purely serial in space whereas Jacobi DDSI is highly paralleliseable. This property means that Jacobi DDSI may be the preferred method for many applications despite displaying a worse rate of convergence in practice.

Our first goal will be to define both DDSI methods to solve the transport equation over a domain with only two subdomains. Once that is achieved we will extend the

methods to work over an arbitrary number of subdomains. However before this we will restate both the transport equation in 3D and the associated source iteration algorithm applied over the whole domain.

4.2.1 Source Iteration Recap

In Section 2.2.1 we defined a simplified version of the neutron transport equation in three spatial dimensions, which was given as

$$\Omega\nabla\cdot\psi(\mathbf{r},\Omega)+\sigma_T(\mathbf{r})\psi(\mathbf{r},\Omega)=\sigma_S(\mathbf{r})\phi(\mathbf{r})+Q(\mathbf{r}), \tag{4.1}$$

for $\mathbf{r}\in V\subset\mathbb{R}^3$ and $\Omega\in\mathbb{S}^2$, subject to

$$\psi(\mathbf{r},\Omega)=f(\mathbf{r},\Omega)\quad\forall\mathbf{r}\in\partial V \text{ such that } \hat{n}(\mathbf{r})\cdot\Omega<0, \tag{4.2}$$

where $\hat{n}(\mathbf{r})$ denotes the outward unit normal vector to V at $\mathbf{r}\in\partial V$, and where

$$\phi(\mathbf{r})\equiv\frac{1}{4\pi}\int_{\mathbb{S}^2}\psi(\mathbf{r},\Omega)\,\mathrm{d}\Omega. \tag{4.3}$$

In Chapter 2 we saw the source iteration algorithm applied to this equation, which we restate here.

Algorithm 4: Source Iteration

1. Start with some initial $\phi^{(0)}(\mathbf{r})$.
2. Solve
$$\Omega\cdot\nabla\psi^{(k+1)}+\sigma_T(\mathbf{r})\psi^{(k+1)}(\mathbf{r},\Omega)=\sigma_S(\mathbf{r})\phi^{(k)}(\mathbf{r})+Q(\mathbf{r}), \tag{4.4}$$
for $\psi^{(k+1)}(\mathbf{r},\Omega)$, subject to
$$\psi^{(k+1)}(\mathbf{r},\Omega)=f(\mathbf{r},\Omega)\quad\forall\mathbf{r}\in\partial V \text{ such that } \hat{n}(\mathbf{r})\cdot\Omega<0. \tag{4.5}$$
3. Average over angle to find
$$\phi^{(k+1)}(\mathbf{r})=\mathcal{P}\psi^{(k+1)}(\mathbf{r},\Omega).$$
4. Increment k and return to step 2.

This method is known to converge, with convergence rate dependent on the scattering ratio (see Section 2.5 for more detail).

Over the next two sections we will present the two different domain decomposition methods talked about in the introduction, namely Jacobi DDSI and Gauss-Seidel DDSI. As mentioned these differ in how they handle passing information across subdomain boundaries and we will both discuss the differences theoretically and observe them numerically after the algorithms have been defined.

4.2.2 Jacobi DDSI

Our aim in this section is to build a domain decomposition source iteration algorithm to work over an arbitrary number of subdomains. This we will refer to as *Jacobi* domain decomposition source iteration (Jacobi DDSI). This algorithm will be focussed towards being highly paralleliseable, but we will see that to achieve this we have had to sacrifice convergence rate to some extent.

We will consider a domain $V \in \mathbb{R}^3$ decomposed into a set of n open, connected and pairwise disjoint subdomains $\{V_i : i = 1, \ldots, n\}$, where $\overline{V} = \bigcup_{i=1}^{n} \overline{V}_i$. However we will first consider the case where $n = 2$ and will build a 2-subdomain version of Jacobi DDSI in Algorithm 5. Using that as a conceptual basis, it will be easy to progress on to an n-subdomain version, given in Algorithm 6.

2-subdomain Jacobi DDSI

We begin by considering the case $n = 2$, so our domain $V \in \mathbb{R}^3$ is decomposed into two subdomains, V_1 and V_2. The intersection, $\partial V_1 \cap \partial V_2$, gives the shared (red, shaded) boundary illustrated in Figure 4-1. To make this easier to refer to, we will use the notation

$$\Gamma_{ij} \equiv \partial V_i \cap \partial V_j.$$

In general this is a 3D surface and not necessarily a plane. This point is worth noting: Jacobi DDSI allows both concave and convex subdomains, meaning the intersection between subdomains need not be a flat surface (or a straight line in 2D). This is an advantage over the Gauss-Seidel DDSI algorithm presented in the next section, and we will expand on this in the discussion at the end of Section 4.2.3.

For each subdomain we need to incorporate boundary conditions along the shared boundary between V_1 and V_2. This can be done in different ways which will form the fundamental difference between Jacobi DDSI and Gauss-Seidel DDSI. In Jacobi iteration we form an intuitive approach in which each updated iteration depends solely upon information from the previous iteration. We will use the notation that, for any function F defined on V, we write

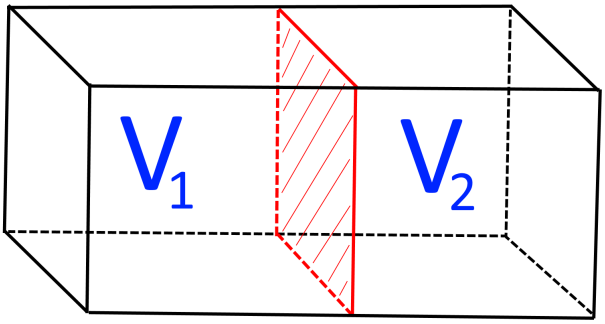

Figure 4-1: Domain V with two subdomains V_1 and V_2.

$$F_i \equiv F\Big|_{\overline{V}_i}, \tag{4.6}$$

where $i = 1, 2$. The DDSI method will iterate over the subdomains, solving for the neutron flux, denoted $\psi_i^{(k+1)}$, in each. To do this we need to impose subdomain-specific boundary conditions that specify the incoming flux along both the subdomain's shared boundary and its external boundary. For each subdomain we split the boundary into two parts: $\partial V_i \cap \partial V$ and $\partial V_i \backslash \partial V$ ($\equiv \Gamma_{12}$ for two subdomains). For $\mathbf{r} \in \partial V_i \cap \partial V$, and for appropriate Ω, we can simply impose the boundary condition, f, as in the full source iteration algorithm. For $\mathbf{r} \in \partial V_i \backslash \partial V$ we will instead match the incoming boundary flux in domain i with the equivalent flux in the neighbouring subdomain at the previous iteration. If we use $\hat{n}_i(\mathbf{r})$ to denote the outward unit normal to ∂V_i at $\mathbf{r} \in \partial V_i$, then for domain V_1 this means that $\forall \mathbf{r} \in \partial V_1$ such that $\hat{n}_1(\mathbf{r}) \cdot \Omega < 0$ we impose

$$\psi_1^{(k+1)}(\mathbf{r}, \Omega) = \begin{cases} \psi_2^{(k)}(\mathbf{r}, \Omega) & \text{when } \mathbf{r} \in \partial V_1 \backslash \partial V \\ f(\mathbf{r}, \Omega) & \text{when } \mathbf{r} \in \partial V_1 \cap \partial V. \end{cases} \tag{4.7}$$

Under this specification we obtain Algorithm 5 given below.

Algorithm 5: Jacobi DDSI - 2 subdomains

1) Start with some initial $\phi_1^{(0)}(\mathbf{r})$ and $\phi_2^{(0)}(\mathbf{r})$.

2) In any order

- Solve

$$\Omega \cdot \nabla \psi_1^{(k+1)} + \sigma_{T1} \psi_1^{(k+1)} = \sigma_{S1} \phi_1^{(k)} + Q_1 \tag{4.8}$$

for $\psi_1^{(k+1)}(\mathbf{r}, \Omega)$ where, $\forall \mathbf{r} \in \partial V_1$ with $\hat{n}_1(\mathbf{r}) \cdot \Omega < 0$, we impose

$$\psi_1^{(k+1)}(\mathbf{r}, \Omega) = \begin{cases} \psi_2^{(k)}(\mathbf{r}, \Omega) & \text{if } \mathbf{r} \in \Gamma_{12}, \\ f(\mathbf{r}, \Omega) & \text{if } \mathbf{r} \in \partial V_1 \cap \partial V. \end{cases}$$

- Solve

$$\Omega \cdot \nabla \psi_2^{(k+1)} + \sigma_{T2} \psi_2^{(k+1)} = \sigma_{S2} \phi_2^{(k)} + Q_2 \tag{4.9}$$

for $\psi_2^{(k+1)}(\mathbf{r}, \Omega)$ where, $\forall \mathbf{r} \in \partial V_2$ with $\hat{n}_2(\mathbf{r}) \cdot \Omega < 0$, we impose

$$\psi_2^{(k+1)}(\mathbf{r}, \Omega) = \begin{cases} \psi_1^{(k)}(\mathbf{r}, \Omega) & \text{if } \mathbf{r} \in \Gamma_{12}, \\ f(\mathbf{r}, \Omega) & \text{if } \mathbf{r} \in \partial V_2 \cap \partial V. \end{cases}$$

3) Average over Ω to find $\phi_1^{(k+1)}(\mathbf{r})$ and $\phi_2^{(k+1)}(\mathbf{r})$ via,

$$\phi_i^{(k+1)}(\mathbf{r}) = \frac{1}{4\pi} \int_{\mathbb{S}^2} \psi_i^{(k+1)}(\mathbf{r}, \Omega) \, \mathrm{d}\Omega, \tag{4.10}$$

for $i = 1, 2$.

4) Increment k and return to step 2.

Though simple, this 2-subdomain Jacobi DDSI algorithm covers all the concepts needed to build an n-subdomain version of Jacobi DDSI.

n-subdomain Jacobi DDSI

We now consider the general decomposition of domain V into n subdomains V_i for $i \in \{1, \ldots, n\}$ as specified earlier. Since any subdomain may border any other subdomain, subdomain-specific boundary conditions in the n-subdomain version of Jacobi DDSI will need to allow for boundary information coming from any other subdomain. This is fairly straightforward, and results in Algorithm 6 below.

Algorithm 6: Jacobi DDSI - n subdomains

1) Choose some initial $\phi_i^{(0)}(\mathbf{r})$ for all $i \in \{1, \ldots, n\}$.

2) For all $i \in \{1, \ldots, n\}$, solve

$$\Omega \cdot \nabla \psi_i^{(k+1)} + \sigma_{Ti} \psi_i^{(k+1)} = \sigma_{Si} \phi_i^{(k)} + Q_i,$$

where, $\forall \mathbf{r} \in \partial V_i$ with $\hat{n}_i(\mathbf{r}) \cdot \Omega < 0$,

$$\psi_i^{(k+1)}(\mathbf{r}, \Omega) = \begin{cases} \psi_j^{(k)}(\mathbf{r}, \Omega) & \text{if } \mathbf{r} \in \Gamma_{ij}, i \neq j, \\ f(\mathbf{r}, \Omega) & \text{if } \mathbf{r} \in \partial V_i \cap \partial V. \end{cases}$$

3) Average over Ω to find $\phi_i^{(k+1)}(\mathbf{r})$ for all $i \in \{1, \ldots, n\}$ via

$$\phi_i^{(k+1)}(\mathbf{r}) = \frac{1}{4\pi} \int_{\mathbb{S}^2} \psi_i^{(k+1)}(\mathbf{r}, \Omega) \, \mathrm{d}\Omega.$$

4) Increment k and return to step 2.

This is the continuous version of the algorithm. In practise we discretise via finite elements and discrete ordinates in order to solve using this algorithm.

So at each iteration, the boundary conditions for each subdomain depend only upon the *previous* iteration in their neighbouring subdomains. This is different to the method of Yavuz and Larsen [77], which is presented with the understanding that each subdomain will be assigned a unique processor. They take advantage of this by taking internal boundary data to be the most recent version of the flux available when it is needed. So for example if, at iteration k, subdomain A needs boundary data from subdomain B, but it turns out that subdomain B has already finished iteration k then the *new* data will be passed to subdomain A. On the other hand, if subdomain B has not yet completed iteration k, then data from the *previous* iteration will be passed to subdomain A. This approach is likely to converge faster than our method since it uses more up-to-date boundary data whenever possible. However our method is presented without assuming any parallelisation, and its consistent internal boundary condition strategy allows for the convergence analysis presented in Section 4.4.

Algorithm 6 easily allows for more complicated geometries, which is a big advantage over the method we will present next. As mentioned it is also very open to parallelisation since each angle can be treated independently, and for each angle we can also solve over each subdomain separately. However, as we will see numerically in Section 4.5, one

downside is that the method converges more slowly than source iteration applied over the whole domain. This may be related to the more restricted communication between the subdomains in the Jacobi version when compared to the Gauss-Seidel version that will be described below.

4.2.3 Gauss-Seidel DDSI

In this section we will build a different domain decomposition source iteration algorithm, which we refer to as Gauss-Seidel DDSI. This algorithm improves upon how the boundary conditions between subdomains were imposed in Jacobi DDSI by more carefully considering the effect of angle on the flow of information through the domain. This improvement comes at the cost of the parallelisation potential that Jacobi DDSI has, and the relative advantages and disadvantages will be discussed fully once the algorithm has been presented.

Similarly to Jacobi DDSI, our goal is to consider decomposing a domain $V \subset \mathbb{R}^3$ into n subdomains V_i for $i \in \{1, \ldots, n\}$. As before we will say that these are open, connected and pairwise disjoint, with $\overline{V} = \bigcup_{i=1}^{n} \overline{V}_i$, however we will also say that each V_i is *convex* which we define here.

Definition 4.1 (Convex [23])**:**
A subset of $\mathbb{R}^n$ *is* convex *if the line segment joining any two points inside it lies wholly inside it.*

Note that convexity of the subdomains was not a requirement for Jacobi DDSI. This is one of the main differences between the two algorithms and will be considered again at the end of this section. We will first consider the case where $n = 2$ and obtain the 2-subdomain Gauss-Seidel DDSI algorithm given in Algorithm 7. From there we will progress to an n-subdomain version in Algorithm 8, and to do so we will have to more carefully consider the order in which we solve the subdomain problems.

2-subdomain Gauss-Seidel DDSI

We start by considering the case $n = 2$, and so have two subdomains V_1 and V_2. Since we require both of them to be convex, the boundary Γ_{12} is necessarily a plane. This allows for the angular domain, $\mathbb{S}^2$, to be separated into three disjoint subdomains:

$$\begin{aligned} U_1 &= \left\{\Omega \in \mathbb{S}^2 : \Omega \cdot \hat{n}_{12} < 0\right\}, \\ U_2 &= \left\{\Omega \in \mathbb{S}^2 : \Omega \cdot \hat{n}_{21} < 0\right\}, \\ U_B &= \left\{\Omega \in \mathbb{S}^2 : \Omega \cdot \hat{n}_{12} = 0 \,(= \Omega \cdot \hat{n}_{21})\right\}, \end{aligned} \tag{4.11}$$

i.e. U_i consists of all angles that are incoming for V_i on the subdomain boundary Γ_{12}, with U_B containing angles parallel to Γ_{12}. Here we have used $\hat{n}_{ij}$ to denote the constant outward unit normal vector from subdomain V_i pointing into V_j (note that $\hat{n}_{12} = -\hat{n}_{21}$).

This is a big difference between Jacobi DDSI and Gauss-Seidel DDSI and will be discussed further at the end of this section. The observation fundamental to this algorithm is then that, for $\Omega \notin U_1$, the angle points from V_1 into V_2. Consequently Γ_{12} is an outflow boundary from V_1, and so the current iterate of the neutron flux on domain V_1 is independent of the current neutron flux iterate on domain V_2. Similarly, for $\Omega \notin U_2$ the current neutron flux iterate on V_2 is independent of the current neutron flux iterate on domain V_1. As a result of this, provided we solve the subdomains in the correct order for each angle, we can always use current iterates of the neutron flux to impose boundary conditions on the internal subdomain boundary. This is in contrast to the Jacobi DDSI algorithm where we always use information from the previous iteration.

To write down an algorithm that uses this approach we have noted that for angles in U_1, solving for ψ_2 on V_2 does not require a boundary condition to be imposed on the internal boundary, Γ_{12}. Similarly for angles in U_2, solving for ψ_1 on V_1 does not require the imposition of a boundary condition on Γ_{12}. Lastly for angles in U_B, neither subdomain requires the imposition of boundary conditions on Γ_{12}. Suppose therefore that for all angles in U_1 we first solve for $\psi_2^{(k+1)}$ on V_2. Then when we come to solve for $\psi_1^{(k+1)}$ on V_1 and need to impose a boundary condition on the internal boundary Γ_{12}, we can use the most up-to-date version of the flux, $\psi_2^{(k+1)}$. In the same manner, if for all angles in U_2 we begin by solving for $\psi_1^{(k+1)}$, then when solving on V_2 we can use $\psi_1^{(k+1)}$ to apply the incoming boundary condition on Γ_{12}. Lastly for all angles in U_B we solve over the subdomains in either order, requiring only the external boundary conditions.

Exploiting the flow of information in this manner leads to the following source iteration algorithm for two subdomains.

Algorithm 7: Gauss-Seidel DDSI - 2 subdomains

1) Start with some initial $\phi_1^{(0)}(\mathbf{r})$ and $\phi_2^{(0)}(\mathbf{r})$.

2)

$\forall \Omega \in U_1$

First solve:
$\mathcal{T}_{\sigma_T}\psi_2^{(k+1)} = \sigma_S\phi_2^{(k)} + Q_2$

subject to:
$\psi_2^{(k+1)}(\mathbf{r},\Omega) = f(\mathbf{r},\Omega)$
$\forall \mathbf{r} \in \partial V$
with $\hat{n}_2(\mathbf{r})\cdot\Omega < 0$

Then solve:
$\mathcal{T}_{\sigma_T}\psi_1^{(k+1)} = \sigma_S\phi_1^{(k)} + Q_1$

subject to:
$$\psi_1^{(k+1)}(\mathbf{r},\Omega) = \begin{cases} \psi_2^{(k+1)}(\mathbf{r},\Omega) & \forall \mathbf{r} \in \Gamma_{12} \\ f(\mathbf{r},\Omega) & \forall \mathbf{r} \in \partial V_1 \cap \partial V \\ & \text{with } \hat{n}_1(\mathbf{r})\cdot\Omega < 0 \end{cases}$$

$\forall \Omega \in U_2$

First solve:
$\mathcal{T}_{\sigma_T}\psi_1^{(k+1)} = \sigma_S\phi_1^{(k)} + Q_1$

subject to:
$\psi_1^{(k+1)}(\mathbf{r},\Omega) = f(\mathbf{r},\Omega)$
$\forall \mathbf{r} \in \partial V$
with $\hat{n}_1(\mathbf{r})\cdot\Omega < 0$

Then solve:
$\mathcal{T}_{\sigma_T}\psi_2^{(k+1)} = \sigma_S\phi_2^{(k)} + Q_2$

subject to:
$$\psi_2^{(k+1)}(\mathbf{r},\Omega) = \begin{cases} \psi_1^{(k+1)}(\mathbf{r},\Omega) & \forall \mathbf{r} \in \Gamma_{12} \\ f(\mathbf{r},\Omega) & \forall \mathbf{r} \in \partial V_2 \cap \partial V \\ & \text{with } \hat{n}_2(\mathbf{r})\cdot\Omega < 0 \end{cases}$$

$\forall \Omega \in U_B$

Solve:
$\mathcal{T}_{\sigma_T}\psi_1^{(k+1)} = \sigma_S\phi_1^{(k)} + Q_1$

subject to:
$\psi_1^{(k+1)}(\mathbf{r},\Omega) = f(\mathbf{r},\Omega)$
$\forall \mathbf{r} \in \partial V$
with $\hat{n}_1(\mathbf{r})\cdot\Omega < 0$

Solve:
$\mathcal{T}_{\sigma_T}\psi_2^{(k+1)} = \sigma_S\phi_2^{(k)} + Q_2$

subject to:
$\psi_2^{(k+1)}(\mathbf{r},\Omega) = f(\mathbf{r},\Omega)$
$\forall \mathbf{r} \in \partial V$
with $\hat{n}_2(\mathbf{r})\cdot\Omega < 0$

3) Average over Ω to find $\phi_1^{(k+1)}(\mathbf{r})$ and $\phi_2^{(k+1)}(\mathbf{r})$ via

$$\phi_j^{(k+1)}(\mathbf{r}) = \frac{1}{4\pi}\int_{\mathbb{S}^2} \psi_j^{(k+1)}(\mathbf{r},\Omega)\,\mathrm{d}\Omega, \qquad j = 1,2.$$

4) Increment k and return to step 2.

Intuitively this presentation of the algorithm makes sense, however to make the step up to n-subdomains we will need to focus more on the *ordering* of the subdomains.

n-subdomain Gauss-Seidel Domain Decomposed Source Iteration

To define the n-subdomain version of Gauss-Seidel DDSI, we will need to order our subdomains in an *appropriate* way for each angle. To define what we mean by this, let us first define an arbitrary ordering of the subdomains using a set O as follows

$$O \equiv \{o_1, \ldots, o_n\},$$

where the order of subdomains is then $V_{o_1}, \ldots, V_{o_n}$.

Definition 4.2 (Appropriate Subdomain Ordering)**:**
For some angle $\Omega \in \mathbb{S}^2$, we define an appropriate *ordering of the subdomains, denoted*

$$O^\Omega \equiv \{o_1, \ldots, o_n\}, \tag{4.12}$$

to be such that any point on the incoming boundary of subdomain V_{o_i} is a point on either the outgoing boundary of V_{o_j} for some $j < i$, or a point on the incoming boundary of the whole domain, ∂V.

The question of whether such an ordering exists for all $\Omega \in \mathbb{S}^2$ for any particular choice of subdomains is important, and we will discuss it further at the end of this section.

For now let us assume that, for each angle $\Omega \in \mathbb{S}^2$, there is an appropriate ordering of the subdomains. Recall that in the 2-subdomain case we broke $\mathbb{S}^2$ into three pieces (U_1, U_2 and U_B) and used these to determine the order in which we solved over our two subdomains. For the n-subdomain algorithm we do not follow this approach. Instead, for each angle $\Omega \in \mathbb{S}^2$, we use the ordering O^Ω to specify the order in which we solve over the subdomains. Because of the way O^Ω is defined we know that every point on the incoming boundary of a subdomain, say V_i, lies either on the domain boundary ∂V or on the outgoing boundary of some previously considered subdomain V_j, $j \neq i$. This means that when imposing incoming boundary conditions for V_i it is always possible to use the current iterate on neighbouring subdomains, or to use the known boundary conditions for the whole domain, V.

With this concept and notation in hand, we can easily write down the n-subdomain version of Gauss-Seidel DDSI as follows.

Algorithm 8: Gauss-Seidel DDSI - n subdomains

1) Choose some initial $\phi_i^{(0)}(\mathbf{r})$ where $\mathbf{r} \in \overline{V}_i$ for all $i \in \{1, \ldots, n\}$.

2) For all $\Omega \in \mathbb{S}^2$, loop over all $i \in O^{\Omega}$ in order, and solve

$$\mathcal{T}_{\sigma_{T_i}} \psi_i^{(k+1)} = \sigma_{Si} \phi_i^{(k)} + Q_i,$$

where, $\forall \mathbf{r} \in \partial V_i$ with $\hat{n}_i(\mathbf{r}) \cdot \Omega < 0$,

$$\psi_i^{(k+1)}(\mathbf{r}, \Omega) = \begin{cases} \psi_j^{(k+1)}(\mathbf{r}, \Omega) & \forall \mathbf{r} \in \Gamma_{ij}, i \neq j, \\ f(\mathbf{r}, \Omega) & \forall \mathbf{r} \in \partial V_i \cap \partial V. \end{cases}$$

3) Average over Ω to find $\phi_i^{(k+1)}(\mathbf{r})$ for all $i \in \{1, \ldots, n\}$ via

$$\phi_i^{(k+1)}(\mathbf{r}) = \frac{1}{4\pi} \int_{\mathbb{S}^2} \psi_i^{(k+1)}(\mathbf{r}, \Omega) \, \mathrm{d}\Omega.$$

4) Increment k and return to step 2.

Algorithm 8 (Gauss-Seidel DDSI) does lend itself to parallelisation, however less so than Algorithm 6 (Jacobi DDSI). The reason for this is that whilst each angle can be considered independently, within each angle the subdomains must be solved in a particular order. In certain cases there may be scope for parallelisation inside each specific subdomain ordering, but this is not easy to see in general. Gauss-Seidel DDSI does have an advantage over Jacobi DDSI in its storage requirements since only the most recent version of the neutron flux needs to be stored at any time. Conversely in Jacobi DDSI the previous iteration must also be stored to facilitate the internal boundary conditions.

As mentioned in Section 1.3, a version of this method is presented by Yavuz and Larsen [77]. They give a semi-discrete algorithm in which the angular variable has already been discretised using discrete ordinates. They assume that the domain is rectangular and has been decomposed into a grid of rectangular subdomains. Then for each angle the subdomains are solved in an appropriate order, as we do in Algorithm 8. In contrast we work with angle as a continuous variable, and allow for the domain to be decomposed in any way provided an appropriate ordering of subdomains exists for every angle, $\Omega \in \mathbb{S}^2$. This requirement is more restrictive than that of Jacobi DDSI for which any choice of decomposition is permissible. It is reasonable to want to decompose the domain into geometrically complicated subdomains motivated by the

physical properties of the problem (as specified by the cross sections) and Jacobi DDSI allows for such generality.

To ensure an appropriate ordering exists in 2D for all $\Omega \in \mathbb{S}^1$ it is sufficient to require only convex subdomains, though concave subdomains can be present under certain conditions (for example on the outer boundary). Whilst we were unable to find a result proving this in the literature, it can be proved using an inductive argument focussing on the overlaps of subdomain boundaries that face the incoming angle (but the proof will not be presented in this thesis). This result is useful since, provided your subdomains are polygonal, it is always possible to further decompose them into a collection of convex subdomains. Similarly we know of no results of this kind for the 3D case. It is possible that our 2D result may also extend to 3D easily, however we have not done this. Practically, for structured grids of convex subdomains appropriate orderings can be easily found on a one-off basis.

We note that it is not required by the algorithm that the whole domain V is convex, only the subdomains. However in Section 4.3 we will prove equivalence of Algorithm 8 to full source iteration applied over the whole domain, V, where the proof requires domain V to be convex.

4.3 Convergence of Gauss-Seidel DDSI

In this section we look at the convergence of Algorithm 8. More precisely, in Theorem 4.4 we will prove that Gauss-Seidel DDSI is equivalent to full source iteration applied over the same domain. As a result it inherits the convergence properties of the well understood source iteration algorithm, which we studied in Chapter 2. This theorem also shows that source iteration can be implemented instead by sweeping through the domain, subdomain by subdomain, solving smaller problems at each iteration. To prove this result we will need to assume that both the whole domain, V, and the subdomains, $V_i \subset V$ with $i \in \{1, \ldots, n\}$, are convex (see Definition 4.1).

To prove our main result we will require Lemma 2.2 and Corollary 2.3 from Chapter 2 concerning the solution to the transport equation under non-zero boundary conditions. We will also need the following lemma linking volume and line integrals.

Lemma 4.3:
Let the domain $V \subset \mathbb{R}^3$ be convex with convex subdomain $W \subset V$. Suppose that, for any point $\mathbf{r} \in W$ and angle $\Omega \in \mathbb{S}^2$, the ray from $\mathbf{r}$ in direction $-\Omega$ intersects ∂W at $\mathbf{r}_1 = \mathbf{r} - s_1\Omega$, and intersects ∂V at $\mathbf{r}_2 = \mathbf{r} - s_2\Omega$ (so $s_2 \geq s_1$, see Figure 4-2). Then

$$\int_{\mathbb{S}^2} \int_{l(\mathbf{r}_1, \mathbf{r}_2)} g(\mathbf{r}') \; dl(\mathbf{r}') \; d\Omega = \int_{V \setminus W} \frac{g(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|_2^2} \; d\mathbf{r}', \tag{4.13}$$

where g is any suitably smooth function. (Note that $\mathbf{r}_i$ and s_i are both functions of $\mathbf{r}$ and Ω for $i = 1, 2$.)

Proof.
The result follows by converting a spherical coordinate integral into a volume integral. We work from the left hand side and by applying the change of variables, $\mathbf{r}' = \mathbf{r} - s'\Omega$, we find

$$\begin{aligned}\int_{\mathbb{S}^2}\int_{l(\mathbf{r}_1,\mathbf{r}_2)} g(\mathbf{r}')\,\mathrm{d}l(\mathbf{r}')\,\mathrm{d}\Omega &= \int_{\mathbb{S}^2}\int_{s_1}^{s_2} g(\mathbf{r}-s'\Omega)\,\mathrm{d}s'\,\mathrm{d}\Omega \\ &= \int_{\mathbb{S}^2}\int_{0}^{s_2} g(\mathbf{r}-s'\Omega)\,\mathrm{d}s'\,\mathrm{d}\Omega - \int_{\mathbb{S}^2}\int_{0}^{s_1} g(\mathbf{r}-s'\Omega)\,\mathrm{d}s'\,\mathrm{d}\Omega.\end{aligned}$$

Now using the volume element $\mathrm{d}V = (s')^2\mathrm{d}s'\mathrm{d}\Omega$, and noting that $s' = \|\mathbf{r}-\mathbf{r}'\|_2$, we obtain

$$\begin{aligned}\int_{\mathbb{S}^2}\int_{l(\mathbf{r}_1,\mathbf{r}_2)} g(\mathbf{r}')\,\mathrm{d}l(\mathbf{r}')\,\mathrm{d}\Omega &= \int_V \frac{g(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|_2^2}\,\mathrm{d}\mathbf{r}' - \int_W \frac{g(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|_2^2}\,\mathrm{d}\mathbf{r}' \\ &= \int_{V\setminus W} \frac{g(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|_2^2}\,\mathrm{d}\mathbf{r}'.\end{aligned}$$

□

In Figure 4-3 we illustrate some new notation that will be instrumental in our next result. We will consider subdomain V_j, for some $j \in \{1, \ldots, n\}$, and will work at a point $\mathbf{r} \in V_j$. From there, a line traced in direction $-\Omega$ crosses the incoming boundary

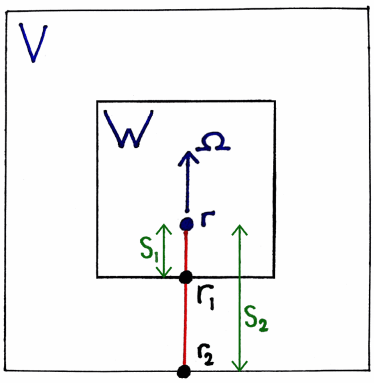

Figure 4-2: Definitions for Lemma 4.3.

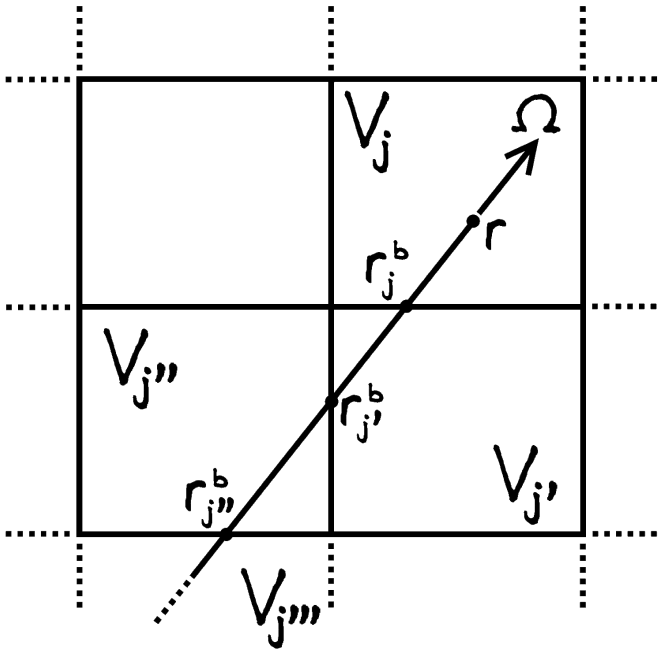

Figure 4-3: Showing a trace back through a domain from pair $(\mathbf{r}, \Omega)$.

of V_j at the point $\mathbf{r}_j^b$. We denote the next subdomain the line passes through by $V_{j'}$, and similarly denote by $\mathbf{r}_{j'}^b$ the point where the line crosses the incoming boundary of $V_{j'}$. We continue this naming convention until the line hits the outer boundary, ∂V.

It is worth noting that the sequence of values, $(j', j'', \dots)$, and also the associated points $\mathbf{r}_\gamma^b$ are dependent upon the pair $(\mathbf{r}, \Omega)$, however we will not usually state this explicitly. We will also recall our earlier notation, $\mathbf{r}^b$, defined in (2.18) as the point lying on the boundary ∂V in direction $-\Omega$ from $\mathbf{r}$, i.e. $\mathbf{r}^b(\mathbf{r}, \Omega) \equiv \mathbf{r} - s^b\Omega$, where

$$s^b(\mathbf{r}, \Omega) \equiv \max\{s \geq 0 : \mathbf{r} - s\Omega \in \partial V\}.$$

With this notation in hand, we can state and prove the main convergence result for Gauss-Seidel DDSI.

Theorem 4.4:
Let the domain $V \subset \mathbb{R}^3$ be convex, and suppose there are n convex, open, connected and pairwise disjoint subdomains V_i for $i \in \{1, \dots, n\}$, such that $\overline{V} = \bigcup_{i=1}^n \overline{V}_i$. Let us assume that for any angle $\Omega \in \mathbb{S}^2$ there is an appropriate ordering of the subdomains (see Definition 4.2). Consider the transport equation as given by

$$\Omega \cdot \nabla\psi(\mathbf{r}, \Omega) + \sigma_T(\mathbf{r})\psi(\mathbf{r}, \Omega) = \sigma_S(\mathbf{r})\phi(\mathbf{r}) + Q(\mathbf{r}), \tag{4.14}$$

with $\mathbf{r} \in V \subset \mathbb{R}^3$ *and* $\Omega \in \mathbb{S}^2$*, subject to boundary conditions*

$$\psi(\mathbf{r}, \Omega) = f(\mathbf{r}, \Omega), \quad \forall \mathbf{r} \in \partial V, \ \ \textit{such that } n(\mathbf{r}) \cdot \Omega < 0, \tag{4.15}$$

where $f \in L^2(\partial V, L^1(\mathbb{S}^2))$ *and we assume* σ_S*,* σ_T *and* Q *are smooth on each subdomain. Solving this using Gauss-Seidel DDSI (Algorithm 8) applied over the* n *subdomains,* $V_1, \ldots, V_n$*, is equivalent to solving using full source iteration (Algorithm 1) applied over the whole domain,* V*.*

Proof.
In this proof (unless otherwise stated) subscript i denotes restriction to subdomain V_i, and we use the convention that

$$\phi^{(k)}(\mathbf{r}) = \phi_i^{(k)}(\mathbf{r}), \tag{4.16}$$

where $\mathbf{r} \in V_i$ for any $i \in \{1, \ldots, n\}$.

We start by choosing an arbitrary subdomain, V_j, where $j \in \{1, \ldots, n\}$. Fix a point $\mathbf{r} \in V_j$ and assume that for all $\Omega \in \mathbb{S}^2$, the line traced from $\mathbf{r}$ in direction $-\Omega$ passes through at least three other subdomains ($V_{j'}$,$V_{j''}$ and $V_{j'''}$) before hitting the outer boundary, ∂V (see Figure 4-3 for an illustration of this).

The majority of this proof will be dedicated to finding an equation for the scalar flux in V_j, and we begin this task now. In Gauss-Seidel DDSI (Algorithm 8) we solve a source problem on V_j with right hand side $\sigma_S \phi_j^{(k)} + Q$, so applying the 3D scalar flux solution, (2.24), we have

$$\phi_j^{(k+1)}(\mathbf{r}) = \int_{\mathbb{S}^2} \psi_{j'}^{(k+1)}(\mathbf{r}_j^b, \Omega) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}_j^b)\right)}{4\pi} \, \mathrm{d}\Omega + \int_{V_j} \left(\sigma_S \phi_j^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}}. \tag{4.17}$$

Next we obtain a formula for the neutron flux term, $\psi_{j'}^{(k+1)}(\mathbf{r}_j^b, \Omega)$, that occurs in the right hand side integral of (4.17). By Algorithm 8 and Lemma 2.2 we have

$$\begin{aligned} \psi_{j'}^{(k+1)}(\mathbf{r}_j^b, \Omega) = \psi_{j''}^{(k+1)}(\mathbf{r}_{j'}^b, \Omega) \exp\left(-\tau(\mathbf{r}_j^b, \mathbf{r}_{j'}^b)\right) \\ + \int_{l(\mathbf{r}_j^b, \mathbf{r}_{j'}^b)} \left(\sigma_S \phi_{j'}^{(k)} + Q\right)(\tilde{\mathbf{r}}) \exp\left(-\tau(\mathbf{r}_j^b, \tilde{\mathbf{r}})\right) \mathrm{d}l(\tilde{\mathbf{r}}). \end{aligned} \tag{4.18}$$

Substituting (4.18) into (4.17) leaves

$$\phi_j^{(k+1)}(\mathbf{r}) = \int_{\mathbb{S}^2} \psi_{j''}^{(k+1)}(\mathbf{r}_{j'}^b, \Omega) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}_j^b) - \tau(\mathbf{r}_j^b, \mathbf{r}_{j'}^b)\right)}{4\pi} \, \mathrm{d}\Omega$$
$$+ \int_{\mathbb{S}^2} \int_{l(\mathbf{r}_j^b, \mathbf{r}_{j'}^b)} \left(\sigma_S \phi_{j'}^{(k)} + Q\right)(\tilde{\mathbf{r}}) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}_j^b) - \tau(\mathbf{r}_j^b, \tilde{\mathbf{r}})\right)}{4\pi} \, \mathrm{d}l(\tilde{\mathbf{r}}) \, \mathrm{d}\Omega$$
$$+ \int_{V_j} \left(\sigma_S \phi_j^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}}. \tag{4.19}$$

Recalling Definition 2.1 of the optical path length, τ, we observe that for any three points $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3 \in \mathbb{R}^3$ satisfying

$$\mathbf{r}_1 + s\Omega = \mathbf{r}_2 + t\Omega = \mathbf{r}_3,$$

with $s > t > 0$ and $\Omega \in \mathbb{S}^2$ fixed, it holds that

$$\begin{aligned}
\tau(\mathbf{r}_1, \mathbf{r}_2) + \tau(\mathbf{r}_2, \mathbf{r}_3) &= \int_{l(\mathbf{r}_1, \mathbf{r}_2)} \sigma_T(\mathbf{z}) \, \mathrm{d}l(\mathbf{z}) + \int_{l(\mathbf{r}_2, \mathbf{r}_3)} \sigma_T(\mathbf{z}) \, \mathrm{d}l(\mathbf{z}) \\
&= \int_{l(\mathbf{r}_1, \mathbf{r}_3)} \sigma_T(\mathbf{z}) \, \mathrm{d}l(\mathbf{z}) \\
&= \tau(\mathbf{r}_1, \mathbf{r}_3).
\end{aligned}$$

This is simply saying that integrating over a line is equivalent to splitting the line into two pieces, and then summing the integrals over those pieces. Consequently we can combine the terms in (4.19) to leave

$$\phi_j^{(k+1)}(\mathbf{r}) = \int_{\mathbb{S}^2} \psi_{j''}^{(k+1)}(\mathbf{r}_{j'}^b, \Omega) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}_{j'}^b)\right)}{4\pi} \, \mathrm{d}\Omega$$
$$+ \int_{\mathbb{S}^2} \int_{l(\mathbf{r}_j^b, \mathbf{r}_{j'}^b)} \left(\sigma_S \phi_{j'}^{(k)} + Q\right)(\tilde{\mathbf{r}}) \frac{\exp\left(-\tau(\mathbf{r}, \tilde{\mathbf{r}})\right)}{4\pi} \, \mathrm{d}l(\tilde{\mathbf{r}}) \, \mathrm{d}\Omega$$
$$+ \int_{V_j} \left(\sigma_S \phi_j^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}}. \tag{4.20}$$

Continuing this process to write $\psi_{j''}^{(k+1)}(\mathbf{r}_{j'}^b, \Omega)$ in terms of information from $V_{j'''}$ in an analagous way, we obtain the formula

$$\phi_j^{(k+1)}(\mathbf{r}) = \int_{\mathbb{S}^2} \psi_{j'''}^{(k+1)}(\mathbf{r}^b_{j''}, \Omega) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}^b_{j''})\right)}{4\pi} \, \mathrm{d}\Omega$$
$$+ \int_{\mathbb{S}^2} \int_{l(\mathbf{r}^b_j, \mathbf{r}^b_{j''})} \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) \frac{\exp\left(-\tau(\mathbf{r}, \tilde{\mathbf{r}})\right)}{4\pi} \, \mathrm{d}l(\tilde{\mathbf{r}}) \, \mathrm{d}\Omega$$
$$+ \int_{V_j} \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}}. \tag{4.21}$$

Here we have dropped the subscript on the iterates, $\phi^{(k)}$, in favour of the combined notation, (4.16). We have then combined two line integrals to obtain the second term on the right hand side of (4.21).

It is clear that we can continue expanding in this manner until we reach the outer boundary at the point $\mathbf{r}^b$. Doing this results in

$$\phi_j^{(k+1)}(\mathbf{r}) = \int_{\mathbb{S}^2} f(\mathbf{r}^b, \Omega) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}^b)\right)}{4\pi} \, \mathrm{d}\Omega$$
$$+ \int_{\mathbb{S}^2} \int_{l(\mathbf{r}^b_j, \mathbf{r}^b)} \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) \frac{\exp\left(-\tau(\mathbf{r}, \tilde{\mathbf{r}})\right)}{4\pi} \, \mathrm{d}l(\tilde{\mathbf{r}}) \, \mathrm{d}\Omega$$
$$+ \int_{V_j} \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}}. \tag{4.22}$$

Directly applying Lemma 4.3 to the second term on the right hand side of (4.22) leaves

$$\int_{\mathbb{S}^2} \int_{l(\mathbf{r}^b_j, \mathbf{r}^b)} \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) \frac{\exp\left(-\tau(\mathbf{r}, \tilde{\mathbf{r}})\right)}{4\pi} \, \mathrm{d}l(\tilde{\mathbf{r}}) \, \mathrm{d}\Omega$$
$$= \int_{V \setminus V_j} \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) \frac{\exp\left(-\tau(\mathbf{r}, \tilde{\mathbf{r}})\right)}{4\pi \left\|\mathbf{r} - \tilde{\mathbf{r}}\right\|_2^2} \, \mathrm{d}\tilde{\mathbf{r}},$$
$$= \int_{V \setminus V_j} \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}},$$

where we have used the definition, (2.32), of the kernel k_{σ_T}. Substituting this back into (4.22) and combining the last two integrals leaves us with a complete formula for $\phi_j^{(k+1)}(\mathbf{r})$, reliant only upon the previous iteration and boundary data,

$$\phi_j^{(k+1)}(\mathbf{r}) = \int_{\mathbb{S}^2} f(\mathbf{r}^b, \Omega) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}^b)\right)}{4\pi} \, \mathrm{d}\Omega + \int_V \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}}. \tag{4.23}$$

Having found this formula, most of the work for this proof has been completed and it just remains to confirm that (4.23) implies Gauss-Seidel DDSI and full SI are equivalent algorithms. To do this we first write down the formula for one iteration of full source iteration, which follows by applying Corollary 2.3 to Algorithm 1 leaving

$$\phi^{(k+1)}(\mathbf{r}) = \int_{\mathbb{S}^2} f(\mathbf{r}^b, \Omega) \frac{\exp\left(-\tau(\mathbf{r}, \mathbf{r}^b)\right)}{4\pi} \, \mathrm{d}\Omega + \int_V \left(\sigma_S \phi^{(k)} + Q\right)(\tilde{\mathbf{r}}) k_{\sigma_T}(\mathbf{r}, \tilde{\mathbf{r}}) \, \mathrm{d}\tilde{\mathbf{r}}. \quad (4.24)$$

Observing that (4.23) and (4.24) are equivalent if we combine the scalar flux as in (4.16) concludes the proof. □

This result is important because it tells us that we can apply source iteration to a domain that has been decomposed into subdomains without any detriment to the rate of convergence. Indeed we are essentially still using full source iteration, however now we have the freedom to treat each subdomain differently. In Chapter 3 we saw how a diffusion equation can be used to accelerate the convergence of source iteration, but also how this acceleration was computationally expensive with respect to the cost of source iteration. Using Gauss-Seidel DDSI we have the ability to build a diffusion-accelerated source iteration algorithm where the acceleration is only applied in specific subdomains. Ideally, these subdomains will be those which exhibit diffusive behaviour and would otherwise display very poor convergence. The potential benefits of this approach will be seen in Section 4.5 where we will look at a 'real-world' 2D model problem.

4.4 Convergence of Jacobi DDSI

In this section we will look at the convergence of the Jacobi DDSI algorithm. Specifically in Theorem 4.8 we will provide a convergence result for Jacobi DDSI with 2 subdomains (Algorithm 5) assuming a convex domain with convex subdomains, and assuming globally constant cross sections. We will prove this by showing that bounds on the error in the iterates satisfy a recurrence relation, which in turn converges under known conditions.

To do this we will need to introduce a new operator, $\mathcal{K}_{ji} : L^2(V_i) \to L^2(V_j)$, see (4.25). This operator is similar to the operator $\mathcal{K}_{\sigma_T}$ defined in Chapter 2 (indeed they are equivalent when $i = j$) however it allows communication between disjoint subdomains. This feature will be invaluable when we return to the convergence of Jacobi DDSI in Section 4.4.2. Before that, we will prove a new bound on the norm of this operator which will be required in the proof of our main convergence result for Jacobi DDSI, Theorem 4.8.

As mentioned, in this section we restrict ourselves to constant material data for our theory. However we note up front that in practice we observe the Jacobi DDSI algorithm converging over all tested ranges of material data and for any tested number of subdomains. This convergence will be seen in our numerical results in Section 4.5, and it indicates that our theory is overly restrictive. Further work and possibly new approaches would hopefully result in a convergence theory that better reflects the observed robustness of the Jacobi DDSI algorithm.

4.4.1 Bounding the Solution Operator Norm

In Chapter 2 we defined the the operator $\mathcal{K}_{\sigma_T} : L^2(V) \to L^2(V)$ where $V \subset \mathbb{R}^k$ for $k = 1, 2$ or 3. Since in this chapter we are splitting the domain into n disjoint subdomains V_i, for $i = 1, \ldots, n$, we can define the related operator $\mathcal{K}_{ji} : L^2(V_i) \to L^2(V_j)$ as

$$\left(\mathcal{K}_{ji} g_i\right)(\mathbf{r}) \equiv \int_{V_i} g_i(\mathbf{r}') k_{\sigma_T}(\mathbf{r}, \mathbf{r}') \, \mathrm{d}\mathbf{r}', \quad \forall \mathbf{r} \in V_j, \tag{4.25}$$

where $g_i \in L^2(V_i)$, for all $i, j \in \{1, \ldots, n\}$. The kernel $k_{\sigma_T}(\mathbf{r}, \mathbf{r}')$ was defined in Chapter 2 for 3D, 2D and 1D domains and assuming piecewise smooth cross sections (see (2.32), (2.34) and (2.36) respectively). Throughout this section we will be assuming the cross sections are globally constant, and so we restate the kernel definitions here in the case of constant cross sections for reference. Firstly for a 3D domain the kernel is defined as

$$k_{\sigma_T}(\mathbf{r}, \mathbf{r}') \equiv \frac{\exp\left(-\sigma_T \left\|\mathbf{r} - \mathbf{r}'\right\|_2\right)}{4\pi \left\|\mathbf{r} - \mathbf{r}'\right\|_2^2}. \tag{4.26}$$

Next, over a 2D domain it is defined as

$$k_{\sigma_T}(\tilde{\mathbf{r}}, \tilde{\mathbf{r}}') \equiv \frac{\exp\left(-\sigma_T \left\|\tilde{\mathbf{r}} - \tilde{\mathbf{r}}'\right\|_2\right)}{2\pi \left\|\tilde{\mathbf{r}} - \tilde{\mathbf{r}}'\right\|_2}. \tag{4.27}$$

Lastly for a 1D domain it is defined to be

$$k_{\sigma_T}(x, x') \equiv \frac{1}{2} E_1(\sigma_T \left|x - x'\right|), \tag{4.28}$$

where E_1 denotes an exponential integral function, defined in [1, p.228, (5.1.4)].

In this section we focus our attention on bounding the norm of the operator $\mathcal{K}_{ji}$ for $i \neq j$. We will prove similar results to Theorems 2.23 and 2.25 (which focussed on the case $i = j$), however by assuming convexity of the subdomains we can obtain tighter bounds. We will see two differences between the bounds in Theorems 4.5 and 4.6 compared to those in Theorems 2.23 and 2.25 respectively. First of all we will

gain a factor of one half in each bound as a result of using only convex subdomains. Secondly, in the exponential power for the 2D and 3D bounds we will have the maximum subdomain diameter ($d_{\max}$) rather than the diameter of a specific subdomain (e.g. $\operatorname{diam}(V_i)$).

We first consider the 2D and 3D cases.

Theorem 4.5:
Let $V \subset \mathbb{R}^k$, $k = 2, 3$, be an open connected set, and let V_1, V_2 be open, connected, convex and disjoint subsets of V such that $\overline{V_1} \cup \overline{V_2} = \overline{V}$. Then the operator $\mathcal{K}_{ji} : L^2(V_i) \to L^2(V_j)$, defined in (4.25) as

$$\left(\mathcal{K}_{ji}\varphi_i\right)(\mathbf{r}) \equiv \int_{V_i} \varphi_i(\mathbf{r}') k_{\sigma_T}(\mathbf{r}, \mathbf{r}') \, d\mathbf{r}', \quad \forall \mathbf{r} \in V_j,$$

for $i, j \in \{1, 2\}, i \neq j$, with $\varphi_i \in L^2(V_i)$ satisfies

$$\|\mathcal{K}_{ji}\|_{L^2(V_i) \to L^2(V_j)} \leq \frac{1}{2\sigma_T} \left(1 - \exp\left(-\sigma_T d_{\max}\right)\right), \tag{4.29}$$

where $d_{\max} \equiv \max_{k=1,2}\{\operatorname{diam}(V_k)\}$, and σ_T is constant.

Proof.
The argument in this proof is similar to that of Theorem 2.23. As in the earlier proof we will work in the case $V \subset \mathbb{R}^3$, with the two dimensional argument being equivalent. First let $\varphi_i \in L^2(V_i)$, then we know

$$\left(\mathcal{K}_{ji}\varphi_i\right)(\mathbf{r}) \equiv \int_{V_i} k_{\sigma_T}(\mathbf{r}, \mathbf{r}')\varphi_i(\mathbf{r}') \, \mathrm{d}\mathbf{r}', \tag{4.30}$$

where $\mathbf{r} \in V_j$, $i \neq j$, and the kernel k_{σ_T} defined as in (4.26). We also know that the kernel satisfies

$$\begin{aligned} &k_{\sigma_T}(\mathbf{r}, \mathbf{r}') > 0, \\ &k_{\sigma_T}(\mathbf{r}, \mathbf{r}') = k_{\sigma_T}(\mathbf{r}', \mathbf{r}), \end{aligned} \tag{4.31}$$

for all $\mathbf{r}, \mathbf{r}' \in V$. Taking the norm of $\mathcal{K}_{ji}\varphi_i$ we find

$$\begin{aligned} \|\mathcal{K}_{ji}\varphi_i\|^2_{L^2(V_j)} &= \int_{V_j} \left| \int_{V_i} k_{\sigma_T}(\mathbf{r}, \mathbf{r}')\varphi_i(\mathbf{r}') \, \mathrm{d}\mathbf{r}' \right|^2 \mathrm{d}\mathbf{r} \\ &= \int_{V_j} \left(\int_{V_i} k_{\sigma_T}(\mathbf{r}, \mathbf{r}')^{1/2} \left|\varphi_i(\mathbf{r}')\right| k_{\sigma_T}(\mathbf{r}, \mathbf{r}')^{1/2} \, \mathrm{d}\mathbf{r}' \right)^2 \mathrm{d}\mathbf{r}, \end{aligned}$$

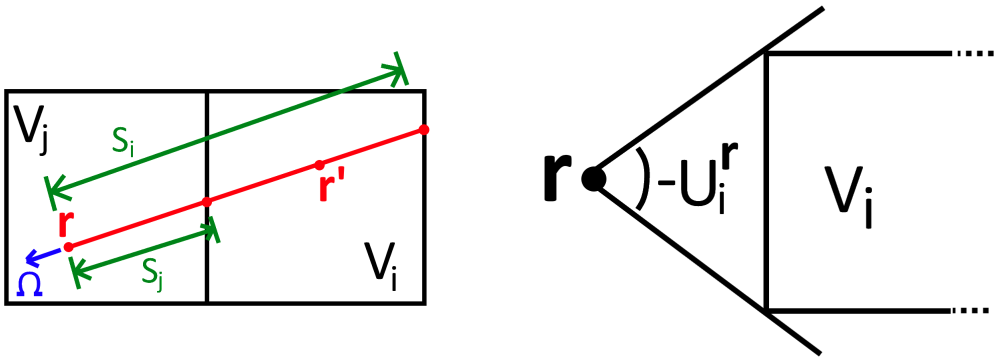

Figure 4-4: Illustrations of the lengths s_i, s_j and of the set of angles $U_i^{\mathbf{r}}$.

then using the Cauchy-Schwarz inequality

$$\begin{aligned}
\|\mathcal{K}_{ji}\varphi_i\|^2_{L^2(V_j)} &\leq \int_{V_j}\left(\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\left|\varphi_i(\mathbf{r}')\right|^2\,\mathrm{d}\mathbf{r}'\right)\left(\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}'\right)\,\mathrm{d}\mathbf{r}\\
&\leq \left(\max_{\mathbf{r}\in V_j}\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}'\right)\int_{V_j}\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\left|\varphi(\mathbf{r}')\right|^2\,\mathrm{d}\mathbf{r}'\,\mathrm{d}\mathbf{r}\\
&= \left(\max_{\mathbf{r}\in V_j}\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}'\right)\int_{V_i}\int_{V_j} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}\left|\varphi(\mathbf{r}')\right|^2\,\mathrm{d}\mathbf{r}'\\
&\leq \left(\max_{\mathbf{r}\in V_j}\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}'\right)\left(\max_{\mathbf{r}'\in V_i}\int_{V_j} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}\right)\|\varphi_i\|^2_{L^2(V_i)}\,. \qquad (4.32)
\end{aligned}$$

Rearranging (4.32) we find

$$\begin{aligned}
\|\mathcal{K}_{ji}\|_{\mathscr{L}(L^2(V_j))} &= \sup_{\substack{\varphi_i\in L^2(V_i)\\ \varphi_i\neq 0}}\frac{\|\mathcal{K}_{ji}\varphi_i\|_{L^2(V_j)}}{\|\varphi_i\|_{L^2(V_i)}}\\
&\leq \left(\max_{\mathbf{r}\in V_j}\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}'\right)^{1/2}\left(\max_{\mathbf{r}'\in V_i}\int_{V_j} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}\right)^{1/2}. \qquad (4.33)
\end{aligned}$$

To get an estimate for (4.33) we first consider

$$\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}',$$

with $\mathbf{r}\in V_j$. We rewrite this in polar coordinates centred at $\mathbf{r}$, so $\mathbf{r}' = \mathbf{r} - s\Omega$, where

$\Omega \in \mathbb{S}^2$ and $s \in \mathbb{R}^+$. Since this gives $\|\mathbf{r} - \mathbf{r}'\|_2 = \|s\Omega\|_2 = s$, recalling (4.26) we now have that

$$k_{\sigma_T}(\mathbf{r}, \mathbf{r}') = \frac{\exp(-\sigma_T s)}{4\pi s^2}. \tag{4.34}$$

We will need some new notation, namely the distance s_j that satisfies,

$$s_j(\mathbf{r}, \Omega) \equiv \max\{s \geq 0 : \mathbf{r} - s\Omega \in \overline{V}_j\}. \tag{4.35}$$

We will also need a new set of angles, denoted $U_i^{\mathbf{r}}$, which contains all the angles $\Omega \in \mathbb{S}^2$ for which a ray, originating at $\mathbf{r} \notin V_i$ and travelling in direction $-\Omega$, will hit any point in V_i. This set is defined, for $\mathbf{r} \in V_i$, as

$$U_i^{\mathbf{r}} \equiv \{\Omega \in \mathbb{S}^2 : \exists s > 0 \text{ for which } \mathbf{r} - s\Omega \in V_i\}. \tag{4.36}$$

These two objects are represented graphically in Figure 4-4. Using them we can write, for $\mathbf{r} \in V_j$,

$$\begin{aligned}\int_{V_i} k_{\sigma_T}(\mathbf{r}, \mathbf{r}')\, \mathrm{d}\mathbf{r}' &= \int_{U_i^{\mathbf{r}}} \int_{s_j}^{s_i} \frac{\exp(-\sigma_T s)}{4\pi s^2} s^2\, \mathrm{d}s\, \mathrm{d}\Omega \\ &= \frac{1}{4\pi} \int_{U_i^{\mathbf{r}}} \int_0^{s_i - s_j} \exp\left(-\sigma_T (s' + s_j)\right) \mathrm{d}s'\, \mathrm{d}\Omega.\end{aligned}$$

Here we have used the change of variables $s' = s - s_j$, and can now just rearrange and integrate

$$\begin{aligned}\int_{V_i} k_{\sigma_T}(\mathbf{r}, \mathbf{r}')\, \mathrm{d}\mathbf{r}' &= \frac{1}{4\pi} \int_{U_i^{\mathbf{r}}} \exp(-\sigma_T s_j) \int_0^{s_i - s_j} \exp\left(-\sigma_T s'\right) \mathrm{d}s'\, \mathrm{d}\Omega \\ &= \frac{1}{4\pi} \int_{U_i^{\mathbf{r}}} \exp(-\sigma_T s_j) \frac{1}{\sigma_T} \underbrace{\left[1 - \exp(-\sigma_T (s_i - s_j))\right]}_{(*)}\, \mathrm{d}\Omega.\end{aligned}$$

We can bound this above by taking the minimum of the distance s_j over angle in the first exponential. We can also bound $(*)$ above by taking the maximum of the distance $s_i - s_j$, which can be taken as the diameter of V_i, i.e. $s_i - s_j \leq \mathrm{diam}(V_i)$. This leaves us with

$$\int_{V_i} k_{\sigma_T}(\mathbf{r}, \mathbf{r}')\, \mathrm{d}\mathbf{r}' \leq \frac{1}{4\pi} \exp\left(-\sigma_T \min_{\Omega \in U_i^{\mathbf{r}}} [s_j(\mathbf{r}, \Omega)]\right) \int_{U_i^{\mathbf{r}}} \frac{1}{\sigma_T} \left[1 - \exp(-\sigma_T \mathrm{diam}(V_i))\right]\, \mathrm{d}\Omega.$$

Next we use the fact that the subdomains are convex to note that $U_i^{\mathbf{r}}$ is at most a hemisphere. Thus the angular integral is bounded above by 2π, and so

$$\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \leq \frac{1}{4\pi}\exp\left(-\sigma_T \min_{\Omega\in U_i^{\mathbf{r}}}\left[s_j(\mathbf{r},\Omega)\right]\right)\frac{2\pi}{\sigma_T}\left[1-\exp\left(-\sigma_T \mathrm{diam}\left(V_i\right)\right)\right].$$

Finally since s_j can potentially be arbitrarily close to zero, we bound the first exponential term above by one, leaving

$$\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \leq \frac{1}{2\sigma_T}\left[1-\exp\left(-\sigma_T \mathrm{diam}\left(V_i\right)\right)\right].$$

Therefore we have found that

$$\max_{\mathbf{r}\in V_j}\int_{V_i} k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' \leq \frac{1}{2\sigma_T}\left[1-\exp\left(-\sigma_T \mathrm{diam}\left(V_i\right)\right)\right], \tag{4.37}$$

for $i \neq j$. Returning to (4.33) we can now conclude

$$\begin{aligned}\left\|\mathcal{K}_{ji}\right\|_{\mathscr{L}(L^2(V_j))} &\leq \left(\frac{1}{2\sigma_T}\left[1-\exp\left(-\sigma_T \mathrm{diam}\left(V_i\right)\right)\right]\right)^{1/2}\left(\frac{1}{2\sigma_T}\left[1-\exp\left(-\sigma_T \mathrm{diam}\left(V_j\right)\right)\right]\right)^{1/2}\\ &\leq \frac{1}{2\sigma_T}\left[1-\exp\left(-\sigma_T d_{\max}\right)\right],\end{aligned}$$

where

$$d_{\max} \equiv \max_{k=1,2}\{\mathrm{diam}\left(V_k\right)\}.$$

□

Next we give an equivalent result for the 1D case.

Theorem 4.6:
Let $V = (x_L, x_R) \subset \mathbb{R}$, and for some $x_M \in V$ define two subsets $V_1 \equiv (x_L, x_M)$ and $V_2 \equiv (x_M, x_R)$ so that $\overline{V_1} \cup \overline{V_2} = \overline{V}$. Then the operator $\mathcal{K}_{ji} : L^2(V_i) \to L^2(V_j)$, defined in (4.25) as

$$\left(\mathcal{K}_{ji}\varphi_i\right)(x) \equiv \int_{V_i} k_{\sigma_T}(x,y)\varphi_i(y)\,dy, \quad \forall x \in V_j,$$

for $i, j \in \{1,2\}, i \neq j$, with kernel k_{σ_T} defined in (4.28) and with $\varphi_i \in L^2(V_i)$, satisfies

$$\left\|\mathcal{K}_{ji}\right\|_{L^2(V_i)\to L^2(V_j)} \leq \frac{1}{2\sigma_T}\left[1-E_2(\sigma_T d_{\max})\right], \tag{4.38}$$

where $d_{\max} \equiv \max\{x_M - x_L, x_R - x_M\}$*,* E_2 *is an* exponential integral *function defined in Abramowitz and Stegun [1, p.228, (5.1.4)], and* σ_T *is constant.*

Proof.
We begin in an equivalent manner as the proof of Theorem 4.5. First let $\varphi_i \in L^2(V_i)$, then we have

$$\left(\mathcal{K}_{ji}\varphi_i\right)(x) \equiv \int_{V_i} k_{\sigma_T}(x,y)\varphi_i(y)\,\mathrm{d}y, \tag{4.39}$$

where $x \in V_j$ and we are always assuming $i, j \in \{1,2\}, i \neq j$, and with

$$k_{\sigma_T}(x,y) \equiv \frac{1}{2}\int_0^1 \frac{1}{\mu}\exp\left(-\frac{\sigma_T}{\mu}\left|x-y\right|\right)\mathrm{d}\mu. \tag{4.40}$$

As in higher dimensions, the kernel satisfies

$$\begin{aligned} &k_{\sigma_T}(x,y) > 0, \\ &k_{\sigma_T}(x,y) = k_{\sigma_T}(y,x), \end{aligned} \tag{4.41}$$

for all $x, y \in V$. Taking the norm of $\mathcal{K}_{ji}\varphi_i$ we can use identical logic as was used to find (4.33), and obtain

$$\left\|\mathcal{K}_{ji}\right\|_{\mathscr{L}(L^2(V_j))} \leq \left(\max_{x\in V_j}\int_{V_i} k_{\sigma_T}(x,y)\,\mathrm{d}y\right)^{1/2}\left(\max_{y\in V_i}\int_{V_j} k_{\sigma_T}(x,y)\,\mathrm{d}x\right)^{1/2}. \tag{4.42}$$

We would now like to estimate the bound in (4.42), and focus to begin with on the maximum

$$\max_{x\in V_1}\int_{V_2} k_{\sigma_T}(x,y)\,\mathrm{d}y. \tag{4.43}$$

We start with the definition of the 1D kernel, (4.40), and use the same change of variables ($x - y = -s\mu$) as was applied to obtain (2.64) to find

$$\begin{aligned} \int_{V_2} k_{\sigma_T}(x,y)\,\mathrm{d}y &= \frac{1}{2}\int_{x_M}^{x_R}\int_0^1 \frac{1}{\mu}\exp\left(-\frac{\sigma_T}{\mu}\left|x-y\right|\right)\mathrm{d}\mu\,\mathrm{d}y, \\ &= \frac{1}{2}\int_0^1 \int_{(x_M-x)/\mu}^{(x_R-x)/\mu} \exp\left(-\sigma_T\left|s\right|\right)\mathrm{d}s\,\mathrm{d}\mu. \end{aligned}$$

Applying the further change of variables $s' = s - (x_M - x)/\mu$ we continue

$$\begin{aligned}\int_{V_2} k_{\sigma_T}(x,y)\,\mathrm{d}y &= \frac{1}{2}\int_0^1 \int_0^{(x_R-x_M)/\mu} \exp\left(-\sigma_T\left(s' + \frac{x_M - x}{\mu}\right)\right)\mathrm{d}s'\,\mathrm{d}\mu \\ &= \frac{1}{2}\int_0^1 \exp\left(-\sigma_T\frac{(x_M-x)}{\mu}\right)\int_0^{(x_R-x_M)/\mu}\exp\left(-\sigma_T s'\right)\mathrm{d}s'\,\mathrm{d}\mu.\end{aligned}$$

Now we can carry out the integral over s', and also (remembering that $x \in V_1 \equiv (x_L, x_M)$) extract the maximum value of $\exp\left(-\sigma_T\frac{(x_M-x)}{\mu}\right)$ over μ, leaving

$$\begin{aligned}\int_{V_2} k_{\sigma_T}(x,y)\,\mathrm{d}y &\leq \frac{1}{2}\exp\left(-\sigma_T(x_M-x)\right)\int_0^1 \frac{1}{\sigma_T}\left[1 - \exp\left(-\sigma_T\frac{(x_R-x_M)}{\mu}\right)\right]\,\mathrm{d}\mu \\ &= \frac{1}{2\sigma_T}\exp\left(-\sigma_T(x_M-x)\right)\left[1 - \int_0^1 \exp\left(-\sigma_T\frac{(x_R-x_M)}{\mu}\right)\mathrm{d}\mu\right].\end{aligned}$$

Lastly we can apply the integral relation (2.59) to be left with

$$\int_{V_2} k_{\sigma_T}(x,y)\,\mathrm{d}y \leq \frac{\exp\left(-\sigma_T(x_M-x)\right)}{2\sigma_T}\big[1 - E_2(\sigma_T(x_R - x_M))\big]. \tag{4.44}$$

Returning now to (4.43) we note that (4.44) is maximised (over $x \in V_1$) at the point $x = x_M$, and so

$$\max_{x\in V_1}\int_{V_2} k_{\sigma_T}(x,y)\,\mathrm{d}y \leq \frac{1}{2\sigma_T}\big[1 - E_2(\sigma_T(x_R - x_M))\big]. \tag{4.45}$$

An identical chain of reasoning yields the complementary bound

$$\max_{y\in V_2}\int_{V_1} k_{\sigma_T}(x,y)\,\mathrm{d}x \leq \frac{1}{2\sigma_T}\big[1 - E_2(\sigma_T(x_M - x_L))\big]. \tag{4.46}$$

Before returning to the operator norm bound (4.42) we make a final observation that both (4.45) and (4.46) are bounded above by

$$\frac{1}{2\sigma_T}\left[1 - E_2(\sigma_T d_{\max})\right],$$

where $d_{\max} \equiv \max\{x_M - x_L, x_R - x_M\}$. This follows from the property given in Remark 2.24 that $E_2(x)$ is decreasing for positive x. Combining this with (4.42) we can thus find that

$$\|\mathcal{K}_{ji}\|_{L^2(V_i)\to L^2(V_j)} \leq \frac{1}{2\sigma_T}\left[1 - E_2(\sigma_T d_{\max})\right],$$

as required. □

This bound is very similar to the bound proved in Theorem 2.25 (for the case $i = j$) but differs by a factor of one half and has $d_{\max}$ where the earlier bound would have $\operatorname{diam}(V_i)$ (previously denoted L). We discussed the earlier result immediately after the proof and much of that discussion is also directly applicable to this new result so will not be repeated. Instead we will simply note once more that $E_2(x)$ is a strictly positive, decreasing function bounded below one for all $x > 0$ (see Remark 2.24).

4.4.2 Convergence of Jacobi DDSI

We will now use the bounds from Section 4.4.1 to prove convergence of Jacobi DDSI in 1D, 2D and 3D under certain assumptions (see Theorem 4.8). We will require knowledge of the convergence of a recurrence relation, which is explained in the following remark.

Remark 4.7:
Consider the recurrence relation

$$x_k = \alpha x_{k-1} + \beta x_{k-2},$$

where $\alpha, \beta > 0$, and suppose we want to find conditions such that

$$\lim_{k\to\infty} x_k = 0. \tag{4.47}$$

The series can be rewritten as

$$\begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} = \begin{pmatrix} \alpha & \beta \\ 1 & 0 \end{pmatrix} \begin{bmatrix} x_{k-1} \\ x_{k-2} \end{bmatrix},$$

and so the series will converge provided the spectral radius of $\left(\begin{smallmatrix} \alpha & \beta \\ 1 & 0 \end{smallmatrix}\right)$ is less than 1. By solving for the eigenvalues of this matrix, we can see that under the conditions $\alpha, \beta > 0$ the spectral radius is less than 1 provided we also have that

$$\alpha + \beta < 1.$$

Now we can state and prove our main convergence result of this section.

Theorem 4.8:
Let $V \subset \mathbb{R}^k$, $k = 1, 2, 3$, be an open, convex domain with subdomains V_1 and V_2 that are open, convex, connected and disjoint, satisfying $\overline{V} = \overline{V}_1 \cup \overline{V}_2$. Then Algorithm 5

(Jacobi DDSI) will converge in 2D and 3D provided

$$\frac{\sigma_S}{\sigma_T}\Big(1-\exp\Big(-\sigma_T d_{\max}\Big)\Big)<\frac{2}{3}, \tag{4.48}$$

and will converge in 1D provided

$$\frac{\sigma_S}{\sigma_T}\Big[1-E_2\Big(\sigma_T d_{\max}\Big)\Big]<\frac{2}{3}, \tag{4.49}$$

where $d_{\max}\equiv\max_{i=1,2}\{\mathrm{diam}\,(V_i)\}$*,* E_2 *is an* exponential integral *function defined in Abramowitz and Stegun [1, p.228, (5.1.4)], and the cross sections are constant.*

Proof.
In this proof we will mostly work in 3D since the lower dimensional argument is a trivial extension. Later in the proof when this is not the case we will make the distinction clear and will present work in both lower and higher dimensions to account for this difference.

In 3D we are using Algorithm 5 to solve the transport equation as given by

$$\Omega\cdot\nabla\psi(\mathbf{r},\Omega)+\sigma_T\psi(\mathbf{r},\Omega)=\sigma_S\phi(\mathbf{r})+Q(\mathbf{r}), \tag{4.50}$$

with $\mathbf{r}\in V\subset\mathbb{R}^3$ and $\Omega\in\mathbb{S}^2$, subject to boundary conditions

$$\psi(\mathbf{r},\Omega)=f(\mathbf{r},\Omega),\quad \forall\mathbf{r}\in\partial V,\ \text{such that}\ \hat{n}(\mathbf{r})\cdot\Omega<0, \tag{4.51}$$

where $f\in L^2(\partial V,L^1(\mathbb{S}^2))$. In this proof (as in Section 4.3) subscript i denotes restriction in space to subdomain V_i, e.g. ϕ_i is the true solution ϕ in domain V_i. Let us denote errors in the neutron flux and scalar flux respectively by

$$\begin{aligned} e_i^{(k)}(\mathbf{r},\Omega)&\equiv\psi_i(\mathbf{r},\Omega)-\psi_i^{(k)}(\mathbf{r},\Omega),\\ E_i^{(k)}(\mathbf{r})&\equiv\phi_i(\mathbf{r})-\phi_i^{(k)}(\mathbf{r}),\end{aligned} \tag{4.52}$$

for $i=1,2$. We first focus on the Jacobi DDSI iterate on subdomain V_1, given by (4.8) in Algorithm 5. Subtracting this from the true transport equation, we can obtain the following equation for the error in subdomain V_1,

$$\Omega\cdot\nabla e_1^{(k+1)}(\mathbf{r},\Omega)+\sigma_T e_1^{(k+1)}(\mathbf{r},\Omega)=\sigma_S E_1^{(k)}(\mathbf{r}), \tag{4.53}$$

where, $\forall\mathbf{r}\in\partial V_1$ with $\hat{n}_1(\mathbf{r})\cdot\Omega<0$, we impose

$$e_1^{(k+1)}(\mathbf{r},\Omega)=\begin{cases} e_2^{(k)}(\mathbf{r},\Omega) & \text{if } \mathbf{r}\in\Gamma_{12},\\ 0 & \text{if } \mathbf{r}\in\partial V_1\cap\partial V.\end{cases}$$

Note that for the error equation the boundary conditions are zero, so Theorems 4.5 and 4.6 are applicable.

Let us define two points, $\mathbf{r}_j(\mathbf{r},\Omega) \equiv \mathbf{r} - s_j\Omega$, where s_j is defined in (4.35). Then using Corollary 2.3 along with the sets $U_i^{\mathbf{r}}$ defined in (4.36) we can find an equation for the scalar flux error,

$$E_1^{(k+1)}(\mathbf{r}) = \int_{U_2^{\mathbf{r}}} e_2^{(k)}(\mathbf{r}_1,\Omega)\frac{\exp\left(-\tau(\mathbf{r},\mathbf{r}_1)\right)}{4\pi}\,\mathrm{d}\Omega + \int_{V_1}\left(\sigma_S E_1^{(k)}\right)(\mathbf{r}')k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}', \quad (4.54)$$

for $\mathbf{r} \in V_1$, where

$$k_{\sigma_T}(\mathbf{r},\mathbf{r}') \equiv \frac{\exp\left(-\sigma_T\left\|\mathbf{r}-\mathbf{r}'\right\|_2\right)}{4\pi\left\|\mathbf{r}-\mathbf{r}'\right\|_2^2}, \quad (4.55)$$

with the optical path length, τ, defined in Definition 2.1. Next we know that $e_2^{(k)}$ satisfies

$$\Omega\cdot\nabla e_2^{(k)}(\mathbf{r},\Omega) + \sigma_T e_2^{(k)}(\mathbf{r},\Omega) = \sigma_S E_2^{(k-1)}(\mathbf{r}),$$

where $\mathbf{r} \in V_2$, and such that, $\forall \mathbf{r} \in \partial V_2$ with $\hat{n}_2(\mathbf{r})\cdot\Omega < 0$,

$$e_2^{(k)}(\mathbf{r},\Omega) = \begin{cases} e_1^{(k-1)}(\mathbf{r},\Omega) & \text{if } \mathbf{r}\in\Gamma_{12}, \\ 0 & \text{if } \mathbf{r}\in\partial V_2\cap\partial V. \end{cases}$$

Therefore, using Lemma 2.2 we know that for $\Omega \in U_2^{\mathbf{r}}$

$$e_2^{(k)}(\mathbf{r}_1,\Omega) = \int_{l(\mathbf{r}_1,\mathbf{r}_2)}\left(\sigma_S E_2^{(k-1)}\right)(\mathbf{r}')\exp\left(-\sigma_T\left\|\mathbf{r}_1-\mathbf{r}'\right\|_2\right)\mathrm{d}\mathbf{r}'. \quad (4.56)$$

If we now substitute this back into (4.54), we can use a simpler version of the argument used in Lemma 4.3 to give

$$E_1^{(k+1)}(\mathbf{r}) = \int_{V_2}\left(\sigma_S E_2^{(k-1)}\right)(\mathbf{r}')k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' + \int_{V_1}\left(\sigma_S E_1^{(k)}\right)(\mathbf{r}')k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}', \quad (4.57)$$

for $\mathbf{r} \in V_1$. An identical chain of reasoning in subdomain V_2 leaves us with

$$E_2^{(k+1)}(\mathbf{r}) = \int_{V_1}\left(\sigma_S E_1^{(k-1)}\right)(\mathbf{r}')k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}' + \int_{V_2}\left(\sigma_S E_2^{(k)}\right)(\mathbf{r}')k_{\sigma_T}(\mathbf{r},\mathbf{r}')\,\mathrm{d}\mathbf{r}', \quad (4.58)$$

for $\mathbf{r} \in V_2$. We have previously defined the operator $\mathcal{K}_{ji} : L^2(V_i) \to L^2(V_j)$ in (4.25) as

$$\left(\mathcal{K}_{ji} g_i\right)(\mathbf{r}) \equiv \int_{V_i} k_{\sigma_T}(\mathbf{r}, \mathbf{r}') g_i(\mathbf{r}')\, \mathrm{d}\mathbf{r}', \quad \forall \mathbf{r} \in V_j. \tag{4.59}$$

Using this we can write $E_1^{(k+1)}$ and $E_2^{(k+1)}$ in operator form as

$$E_1^{(k+1)}(\mathbf{r}) = \mathcal{K}_{11}\left(\sigma_S E_1^{(k)}\right) + \mathcal{K}_{12}\left(\sigma_S E_2^{(k-1)}\right),$$

$$E_2^{(k+1)}(\mathbf{r}) = \mathcal{K}_{22}\left(\sigma_S E_2^{(k)}\right) + \mathcal{K}_{21}\left(\sigma_S E_1^{(k-1)}\right).$$

Combining these we can obtain the following operator-matrix equation

$$\begin{bmatrix} E_1^{(k+1)} \\ E_2^{(k+1)} \end{bmatrix} = \begin{bmatrix} \sigma_S\mathcal{K}_{11} & 0 \\ 0 & \sigma_S\mathcal{K}_{22} \end{bmatrix} \begin{bmatrix} E_1^{(k)} \\ E_2^{(k)} \end{bmatrix} + \begin{bmatrix} 0 & \sigma_S\mathcal{K}_{12} \\ \sigma_S\mathcal{K}_{21} & 0 \end{bmatrix} \begin{bmatrix} E_1^{(k-1)} \\ E_2^{(k-1)} \end{bmatrix}. \tag{4.60}$$

Going forward, for any $x \in L^2(V_1)$ and $y \in L^2(V_2)$, we will use the following norm notation

$$\left\| \begin{matrix} x \\ y \end{matrix} \right\|_2 \equiv \left\| \begin{matrix} x \\ y \end{matrix} \right\|_{L^2(V_1)\times L^2(V_2)} = \sqrt{\|x\|^2_{L^2(V_1)} + \|y\|^2_{L^2(V_2)}}.$$

Furthermore we will extend the operator norm notation introduced in Section 2.2.4 so that for any operators $\mathcal{A}_{ji} : L^2(V_i) \to L^2(V_j)$ we write $\|\mathcal{A}_{ji}\|_{\mathscr{L}_{ji}} \equiv \|\mathcal{A}_{ji}\|_{\mathscr{L}(L^2(V_i)\to L^2(V_j))}$. Lastly, for 2x2 operator matrices we specify the norm

$$\left\| \begin{matrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{matrix} \right\|_{\mathscr{L}} \equiv \sup_{\left\| \begin{matrix} x \\ y \end{matrix} \right\|_2 = 1} \left\| \begin{matrix} \mathcal{A}_{11}x + \mathcal{A}_{12}y \\ \mathcal{A}_{21}x + \mathcal{A}_{22}y \end{matrix} \right\|_2.$$

Using these we see that

$$\begin{aligned}
\left\| \begin{array}{cc} 0 & \mathcal{A}_{12} \\ \mathcal{A}_{21} & 0 \end{array} \right\|_{\mathscr{L}} &= \sup_{\left\| \begin{smallmatrix} x \\ y \end{smallmatrix} \right\|_2=1} \left\| \begin{array}{c} \mathcal{A}_{12}y \\ \mathcal{A}_{21}x \end{array} \right\|_2, \\
&= \sup_{\left\| \begin{smallmatrix} x \\ y \end{smallmatrix} \right\|_2=1} \sqrt{\|\mathcal{A}_{12}y\|^2_{L^2(V_1)} + \|\mathcal{A}_{21}x\|^2_{L^2(V_2)}} \\
&\leq \sup_{\left\| \begin{smallmatrix} x \\ y \end{smallmatrix} \right\|_2=1} \sqrt{\|\mathcal{A}_{12}\|^2_{\mathscr{L}_{12}} \|y\|^2_{L^2(V_2)} + \|\mathcal{A}_{21}\|^2_{\mathscr{L}_{21}} \|x\|^2_{L^2(V_1)}} \\
&\leq \sup_{\left\| \begin{smallmatrix} x \\ y \end{smallmatrix} \right\|_2=1} \left[\max\left\{ \|\mathcal{A}_{12}\|_{\mathscr{L}_{12}}, \|\mathcal{A}_{21}\|_{\mathscr{L}_{21}} \right\} \sqrt{\|y\|^2_{L^2(V_2)} + \|x\|^2_{L^2(V_1)}} \right] \\
&= \max\left\{ \|\mathcal{A}_{12}\|_{\mathscr{L}_{12}}, \|\mathcal{A}_{21}\|_{\mathscr{L}_{21}} \right\}. \qquad (4.61)
\end{aligned}$$

Using similar logic we also know that

$$\left\| \begin{array}{cc} \mathcal{A}_{11} & 0 \\ 0 & \mathcal{A}_{22} \end{array} \right\|_{\mathscr{L}} \leq \max\left\{ \|\mathcal{A}_{11}\|_{\mathscr{L}_{11}}, \|\mathcal{A}_{22}\|_{\mathscr{L}_{22}} \right\}. \qquad (4.62)$$

Continuing now with the proof, we take norms of (4.60) and use the triangle inequality to get

$$\left\| \begin{array}{c} E_1^{(k+1)} \\ E_2^{(k+1)} \end{array} \right\|_2 \leq \left\| \begin{bmatrix} \sigma_S\mathcal{K}_{11} & 0 \\ 0 & \sigma_S\mathcal{K}_{22} \end{bmatrix} \begin{bmatrix} E_1^{(k)} \\ E_2^{(k)} \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} 0 & \sigma_S\mathcal{K}_{12} \\ \sigma_S\mathcal{K}_{21} & 0 \end{bmatrix} \begin{bmatrix} E_1^{(k-1)} \\ E_2^{(k-1)} \end{bmatrix} \right\|_2,$$

and so

$$\left\| \begin{array}{c} E_1^{(k+1)} \\ E_2^{(k+1)} \end{array} \right\|_2 \leq \left\| \begin{array}{cc} \sigma_S\mathcal{K}_{11} & 0 \\ 0 & \sigma_S\mathcal{K}_{22} \end{array} \right\|_{\mathscr{L}} \left\| \begin{array}{c} E_1^{(k)} \\ E_2^{(k)} \end{array} \right\|_2 + \left\| \begin{array}{cc} 0 & \sigma_S\mathcal{K}_{12} \\ \sigma_S\mathcal{K}_{21} & 0 \end{array} \right\|_{\mathscr{L}} \left\| \begin{array}{c} E_1^{(k-1)} \\ E_2^{(k-1)} \end{array} \right\|_2. \qquad (4.63)$$

At this point the result in 1D varies from that in 2D and 3D. We will focus on the 2D and 3D cases together first, returning to the 1D case afterwards.

Applying (4.61) and using the bound from Theorem 4.5 we have that

$$\begin{aligned}
\left\| \begin{array}{cc} 0 & \sigma_S\mathcal{K}_{12} \\ \sigma_S\mathcal{K}_{21} & 0 \end{array} \right\|_{\mathscr{L}} &\leq \max_{i \neq j} \left\{ \sigma_S \|\mathcal{K}_{ij}\|_{\mathscr{L}_{ij}} \right\}, \\
&\leq \frac{\sigma_S}{2\sigma_T} \left(1 - \exp\left(-\sigma_T d_{\max} \right) \right),
\end{aligned}$$

where we have used the notation $d_{\max} \equiv \max_{i=1,2}\{\text{diam}\,(V_i)\}$. In a similar way, applying (4.62) and using the norm bound in Theorem 2.23, we can say

$$\begin{aligned}
\left\| \begin{array}{cc} \sigma_S \mathcal{K}_{11} & 0 \\ 0 & \sigma_S \mathcal{K}_{22} \end{array} \right\|_{\mathscr{L}} &\leq \max_i \left\{ \sigma_S \left\| \mathcal{K}_{ii} \right\|_{\mathscr{L}_{ii}} \right\}, \\
&\leq \frac{\sigma_S}{\sigma_T} \max_i \left\{ 1 - \exp\left(-\sigma_T \text{diam}\,(V_i) \right) \right\} \\
&\leq \frac{\sigma_S}{\sigma_T} \left(1 - \exp\Big(-\sigma_T d_{\max} \Big) \right).
\end{aligned}$$

We now note that (4.63) is simply a recurrence relation, and to simplify the expression we define

$$\begin{aligned}
\alpha &\equiv \frac{\sigma_S}{\sigma_T} \left(1 - \exp\Big(-\sigma_T d_{\max} \Big) \right), \\
\beta &\equiv \frac{\sigma_S}{2\sigma_T} \left(1 - \exp\Big(-\sigma_T d_{\max} \Big) \right), \\
v_k &\equiv \left\| \begin{array}{c} E_1^{(k)} \\ E_2^{(k)} \end{array} \right\|_2 .
\end{aligned}$$

Then from (4.63) we have that

$$v_{k+1} \leq \alpha v_k + \beta v_{k-1}. \tag{4.64}$$

We know from Remark 4.7 that the associated recurrence relation

$$x_{k+1} = \alpha x_k + \beta x_{k-1}, \tag{4.65}$$

satisfies

$$\lim_{k\to\infty} x_k = 0, \quad \forall \alpha, \beta \in \mathbb{R}^+ \text{ such that } \alpha + \beta < 1. \tag{4.66}$$

Fixing $x_1 = v_1$, $x_2 = v_2$, we have that $v_k \leq x_k$ for all $k \geq 3$. Thus we can say that

$$\lim_{k\to\infty} v_k = 0, \quad \text{provided } \alpha + \beta < 1.$$

Therefore our condition for convergence can be resolved to

$$\alpha + \beta < 1$$

$$\Rightarrow \qquad \frac{\sigma_S}{\sigma_T}\Big(1 - \exp\Big(-\sigma_T d_{\max}\Big)\Big) < \frac{2}{3}. \tag{4.67}$$

This is the condition we were looking for in 2D and 3D, however in 1D the result is a little different and we will tackle this case now.

Similarly to the 2D and 3D case, we can use the norm bounds in Theorem 2.25 and Theorem 4.6 to say

$$\begin{aligned}
\left\| \begin{array}{cc} \sigma_S\mathcal{K}_{11} & 0 \\ 0 & \sigma_S\mathcal{K}_{22} \end{array} \right\|_{\mathscr{L}} &\leq \frac{\sigma_S}{\sigma_T} \max_i \Big\{1 - E_2\Big(\sigma_T \mathrm{diam}\,(V_i)\Big)\Big\} \\
&\leq \frac{\sigma_S}{\sigma_T}\Big[1 - E_2\Big(\sigma_T d_{\max}\Big)\Big], \qquad (\star)
\end{aligned}$$

$$\left\| \begin{array}{cc} 0 & \sigma_S\mathcal{K}_{12} \\ \sigma_S\mathcal{K}_{21} & 0 \end{array} \right\|_{\mathscr{L}} \leq \frac{\sigma_S}{2\sigma_T}\Big[1 - E_2\Big(\sigma_T d_{\max}\Big)\Big],$$

where in obtaining $(\star)$ we have referred to Remark 2.24, allowing us to note that $1 - E_2(x)$ is increasing for positive x. Once again we can reformulate (4.63) as a recurrence relation, $v_{k+1} \leq \hat{\alpha} v_k + \hat{\beta} v_{k-1}$, where this time

$$\hat{\alpha} \equiv \frac{\sigma_S}{\sigma_T}\Big[1 - E_2\Big(\sigma_T d_{\max}\Big)\Big],$$

$$\hat{\beta} \equiv \frac{\sigma_S}{2\sigma_T}\Big[1 - E_2\Big(\sigma_T d_{\max}\Big)\Big].$$

Lastly using Remark 4.7 we find the 1D condition for convergence to be

$$\hat{\alpha} + \hat{\beta} < 1$$

$$\Rightarrow \qquad \frac{\sigma_S}{\sigma_T}\Big[1 - E_2\Big(\sigma_T d_{\max}\Big)\Big] < \frac{2}{3}, \tag{4.68}$$

as required. □

An immediate consequence of this theorem is that Jacobi DDSI over two convex

subdomains converges provided the scattering ratio in each subdomain is less than 2/3. Taking slightly more care we can show that in 2D and 3D Jacobi DDSI over two convex subdomains will converge provided

$$d_{\max} < \frac{1}{\sigma_T} \log_e(3).$$

Whilst the conditions in Theorem 4.8 do guarantee convergence of the method, they are overly strict and in practice the method converges for all tested ranges of scattering ratio for problems with 2 or more subdomains. Indeed, in Section 4.5 we will explore the convergence of 2D Jacobi DDSI numerically and will demonstrate convergence over both a larger range of scattering ratios than predicted by our theory, and also for more than two subdomains (see the tests in Section 4.5.2). A proof that reflects this observed robustness of the method has not yet been found.

4.5 Numerical Tests

In this section we will present results from several different numerical experiments. As in our last two numerical results sections, we consider the problem of numerically solving the neutron transport equation, subject to zero incoming boundary conditions, for a 2D spatial domain and a 1D angular domain. As explained in Section 3.5, to carry out these tests we first discretised the neutron transport equation using discontinuous Galerkin finite elements in space (see [61], [42]) and discrete ordinates in angle (see [22], [32, Chapter 9]). A detailed account of this discretisation is given in Chapter 5, and knowledge of that chapter will be assumed at times during this section. We also discretised the diffusion equation using continuous finite elements, though we don't cover the continuous discretisation in detail. For information on this, see for example Brenner and Scott [16].

In the first experiments we will numerically confirm our convergence results for Gauss-Seidel DDSI and Jacobi DDSI: Theorem 4.4 and Theorem 4.8 respectively. We will see that our results do hold numerically, and in particular that Jacobi DDSI converges under far fewer assumptions than Theorem 4.8 requires. To conclude the chapter we will show the effectiveness of applying DSA only in subdomains for which it will have a big effect. To do this we will implement a physically motivated example of a nuclear fuel storage pool. We will see that this 'hybrid' approach allows us to obtain the benefits of DSA while limiting the added computational cost.

4.5.1 Verifying Jacobi DDSI Convergence

In this first section we will focus just on numerically confirming the conclusion of Theorem 4.8, concerning the convergence of Jacobi DDSI (Algorithm 6). This result proved that 2D Jacobi DDSI applied over a convex domain decomposed into two convex, connected and pairwise disjoint subdomains will converge provided

$$\frac{3}{2}\frac{\sigma_S}{\sigma_T}\left(1-\exp\Big(-\sigma_T d_{\max}\Big)\right)<1, \tag{4.69}$$

where $d_{\max} \equiv \max_{i=1,2}\{\text{diam}\,(V_i)\}$. Therefore, in our numerical tests when this inequality is satisfied we expect Jacobi DDSI to converge, and we should measure an error ratio less than 1. While this bound will be seen to guarantee convergence, we will also see that it is overly pessimistic. In Section 4.5.2 we will demonstrate that Jacobi DDSI converges for a much wider range of problems than we test here.

To test Theorem 4.8 we will solve the transport equation over a square domain decomposed into two subdomains, and will try varying two quantities separately. Firstly we will repeat our numerical test in Section 2.7.1 by varying the scattering ratio σ_S/σ_T, and secondly we will vary the domain diameter but maintain the domain's square shape.

So first of all we set our domain, $V = [0,1]\times[0,1]$, and decompose it into two subdomains: $V_1 = [0,0.5]\times[0,1]$ and $V_2 = [0.5,1]\times[0,1]$. We set the source, Q, and total cross section, σ_T, to be 1. Similarly to our numerical test in Section 2.7.1, we will vary σ_A from 0.9 down to -1.5, causing the scattering ratio, σ_S/σ_T, to range from 0.1 up to 2.5. Whilst this range of values is not physically realistic, as mentioned in Section 2.7 it can be interpreted as a very basic inclusion of fission and is also theoretically interesting. For each value we run Jacobi DDSI until it converges to a tolerance of 10^{-4} or reaches 25 iterations (whichever occurs sooner). We use an initial guess of zero, with spatial resolutions $M_x = M_y = 16$ and angular resolution $N = 32$.

In Table 4.1 for each value of the scattering ratio (column 1) we have given the observed error ratio of Jacobi DDSI (column 2) and the value of the left hand side of the convergence condition, (4.69), which implies convergence for continuous Jacobi DDSI (column 3).

By comparing columns 2 and 3 it is clear that these results support the conclusion of Theorem 4.8, since Jacobi DDSI converges for all tested values of the scattering ratio. It is also clear that the criteria (4.69) is not strict since (in this test at least) for values of the scattering ratio over roughly one it does not hold, yet Jacobi DDSI still converges. This result is not unexpected since to derive the convergence criteria we took several inequalities and imposed several strict assumptions, so there is definitely scope for improvement.

Scattering ratio, σ_S/σ_T	Observed Error Ratio of Jacobi DDSI	Convergence criteria: left side of (4.69)
0.1	0.075	0.101
0.3	0.204	0.303
0.5	0.297	0.505
0.7	0.373	0.707
0.9	0.440	0.909
1.1	0.505	1.111
1.3	0.568	1.313
1.5	0.629	1.514
1.7	0.689	1.716
1.9	0.748	1.918
2.1	0.807	2.120
2.3	0.865	2.322
2.5	0.922	2.524

Table 4.1: Table of the observed error ratio of Jacobi DDSI and the left hand side of convergence criteria (4.69) for varying scattering ratio.

For the second test we set our domain, $V = [0, D] \times [0, D]$, where $D \in \mathbb{R}^+$ will be specified, and decompose it into two subdomains: $V_1 = [0, D/2] \times [0, D]$ and $V_2 = [D/2, D] \times [0, D]$. We note that for values of the scattering ratio less than $2/3$, the inequality (4.69) will always be satisfied regardless of the domain size (since the exponential is always positive). For this test we will fix the scattering ratio to be $2/3$, however we will discuss other choices afterwards. This means the left hand side of (4.69) will approach 1 as D increases.

We set the source, Q, and total cross section, σ_T, to be 1, and set the absorption cross section, σ_A to be $1/3$ (resulting in $\sigma_S/\sigma_T = 2/3$). We then vary the parameter D between 10^{-2} and 10^4, causing $d_{\max}$ to vary also since

$$d_{\max} = \sqrt{D^2 + (D/2)^2} = D\sqrt{5}/2.$$

For each value we run Jacobi DDSI until it converges to a tolerance of 10^{-8} or reaches 25 iterations (whichever occurs sooner). We use an initial guess of zero, with spatial resolutions $M_x = M_y = 16$ and angular resolution $N = 32$. Due to the available hardware it was not feasible to increase the resolution beyond this point and so the mesh width for wider domains is very large. Consequently we have not included these results in Table 4.2 since they are likely to be inaccurate, however we have included them in Figure 4-5 for illustration.

In Table 4.2 for each value of the domain size parameter, D, (column 1) we have

Domain width and height, D	Max. subdomain diameter, $d_{\max} = D\sqrt{5}/2$	Jacobi DDSI Error Ratio	Convergence criteria, (4.69)
$10^{-1.0}$	0.11	0.151	0.115
$10^{-0.5}$	0.35	0.199	0.321
$10^{0.0}$	1.12	0.360	0.706
$10^{0.5}$	3.54	0.551	0.979
$10^{1.0}$	11.18	0.661	1.000
$10^{1.5}$	35.36	0.681	1.000
$10^{2.0}$	111.80	0.673	1.000

Table 4.2: Table giving the observed error ratio of Jacobi DDSI when applied over two subdomains, as well as the left hand side of convergence criteria (4.69) for varying domain width.

given the resulting value of the maximum subdomain diameter, $d_{\max}$, (column 2) along with the observed error ratio of Jacobi DDSI (column 3) and the value of the left hand side of the convergence condition, (4.69), which implies convergence for continuous Jacobi DDSI when below one (column 4).

Looking at columns 3 and 4 in Table 4.2 it is immediately clear that our theory is supported since Jacobi DDSI converges over the whole tested range of diameters. Interestingly we see that actual observed error ratio peaks at about 0.681 before settling down to 2/3 (the scattering ratio). This is also clear visually in Figure 4-5, which plots the data from column 3 versus $d_{\max}$. The peak suggests that taking the scattering ratio close enough to 1 might induce divergence of the method, however in practice this does not occur and instead the peak reduces in size enough to not cause divergence.

By fixing the scattering ratio above 2/3 we can cause the criteria (4.69) to be violated for large enough $d_{\max}$. However for values of the scattering ratio less than 1 we have always found Jacobi DDSI converges. Consequently we note (as we did in the first test) that our convergence criteria appears to be overly strict and that improvements in the theory are likely to exist.

4.5.2 Verifying Gauss-Seidel DDSI Convergence

Our main focus in this section is to numerically support Theorem 4.4: our convergence result concerning Gauss-Seidel DDSI (Algorithm 8). However, each test will also be carried out on Jacobi DDSI in order to further demonstrate that it converges for a wide range of domain decompositions and material data. We will see both methods converge in all situations that we test.

We will repeat our earlier experiment on full SI from Section 2.7.1 in which we

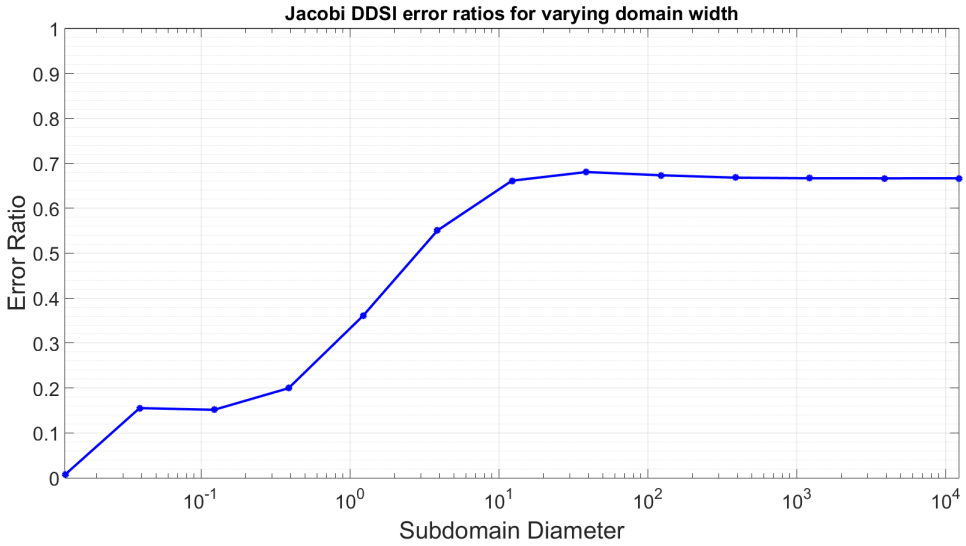

Figure 4-5: Figure showing the observed error ratio of Jacobi DDSI for varying subdomain diameter, $d_{\max}$.

solved the transport equation over a square domain, $V = [0, 1] \times [0, 1]$, whilst varying the scattering ratio over the whole domain. However, as an extension to this we will also vary the number of subdomains that V is decomposed into.

We set the source, Q, and total cross section, σ_T, to be 1, then we vary σ_A from 0.9 down to -1.5, causing the scattering ratio, σ_S/σ_T, to vary from 0.1 up to 2.5. For each value we run full source iteration as well as both DDSI algorithms until they converge to a tolerance of 10^{-4}. We use an initial guess of zero, with spatial resolutions $M_x = M_y = 8$ and angular resolution $N = 23$.

As mentioned, we will also vary the number of subdomains that the DDSI algorithms are using, starting with just the square domain, V. We will then divide V into a 2x2 grid of 4 square subdomains, before next subdividing each of these subdomains to yield a 4x4 square grid (16 subdomains) and then lastly an 8x8 grid (64 subdomains). The first two refinements are illustrated in Figure 4-6, with the 2x2 grid depicted with solid lines, and the 4x4 grid with the extra dotted lines. For each level of refinement we will run the full test varying the scattering ratio for each of our DDSI algorithms, and will then look at how higher numbers of subdomains impacts the convergence of each of our DDSI algorithms.

Before presenting the results we will consider what we should expect to see. In Theorem 4.4 we proved that applying Gauss-Seidel DDSI (Algorithm 8) over a convex domain decomposed into a finite number of convex, connected and pairwise disjoint subdomains is equivalent to applying full source iteration (Algorithm 1) over the whole

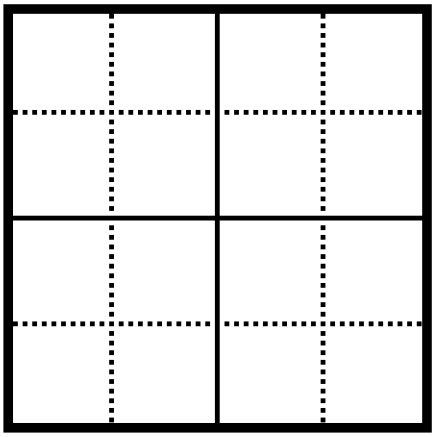

Figure 4-6: Illustration of two successive refinements of subdomains: 1x1 (Bold, outer lines), 2x2 (solid, internal lines) and 4x4 (dotted internal lines).

domain. We would like to observe this equivalence numerically, and so expect to see identical convergence rates from full SI and Gauss-Seidel DDSI.

In Table 4.3 for each value of the scattering ratio (column 1) we have given the error ratios for full SI (column 2), Gauss-Seidel DDSI (column 3), and Jacobi DDSI (column 4). These data are for solves over domain V decomposed into a 4x4 grid of 16 subdomains.

Looking first just at columns 2 and 3 we can see that full source iteration and Gauss-Seidel DDSI do appear to converge identically, supporting the conclusion of Theorem 4.4. The differences between columns 2 and 3 are in fact of $\mathcal{O}(10^{-13})$, and they remain of this order for any tested number of subdomains greater than 1. The small difference is most likely due to rounding error within the computation. When only 1 subdomain is used, both of our DDSI algorithms are indistinguishable from full source iteration to machine precision, as we would expect.

Next considering the data in column 4 we can see that Jacobi DDSI (applied over the square grid of 16 subdomains) also converges over the whole range of tested scattering ratios, albeit slightly more slowly than the other algorithms. This further demonstrates the robustness we have observed in this method.

Figure 4-7 contains a plot of the data in Table 4.3 and shows this slower convergence rate. However it also shows that the error ratios of the three methods get closer together as the scattering ratio increases. In fact, by trial and error we found that for a scattering ratio of c.2.775 the three methods all have the same error ratio and begin to diverge.

The slower convergence of Jacobi DDSI stems from how the method handles data crossing subdomain boundaries. Gauss-Seidel DDSI carefully solves in such a way

Scattering ratio, σ_S/σ_T	Observed Error Ratio for full SI	Gauss-Seidel DDSI	Jacobi DDSI
0.1	0.036	0.036	0.390
0.3	0.108	0.108	0.476
0.5	0.180	0.180	0.549
0.7	0.252	0.252	0.605
0.9	0.324	0.324	0.655
1.1	0.396	0.396	0.700
1.3	0.468	0.468	0.742
1.5	0.541	0.541	0.781
1.7	0.613	0.613	0.819
1.9	0.685	0.685	0.855
2.1	0.757	0.757	0.890
2.3	0.829	0.829	0.923
2.5	0.901	0.901	0.956

Table 4.3: Table giving the observed error ratio for full SI, Gauss-Seidel DDSI and Jacobi DDSI for varying scattering ratios, when used to solve over a domain V decomposed into a 4x4 grid of 16 subdomains.

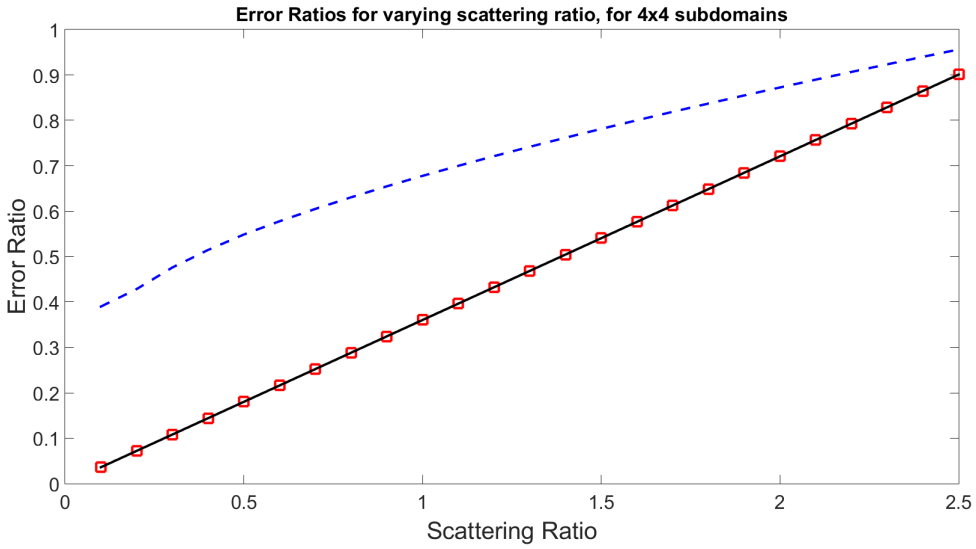

Figure 4-7: Plot of the observed error ratio versus scattering ratio for Jacobi DDSI (blue, dashed line), Gauss-Seidel DDSI (red boxes, no line) and full SI (black solid line).

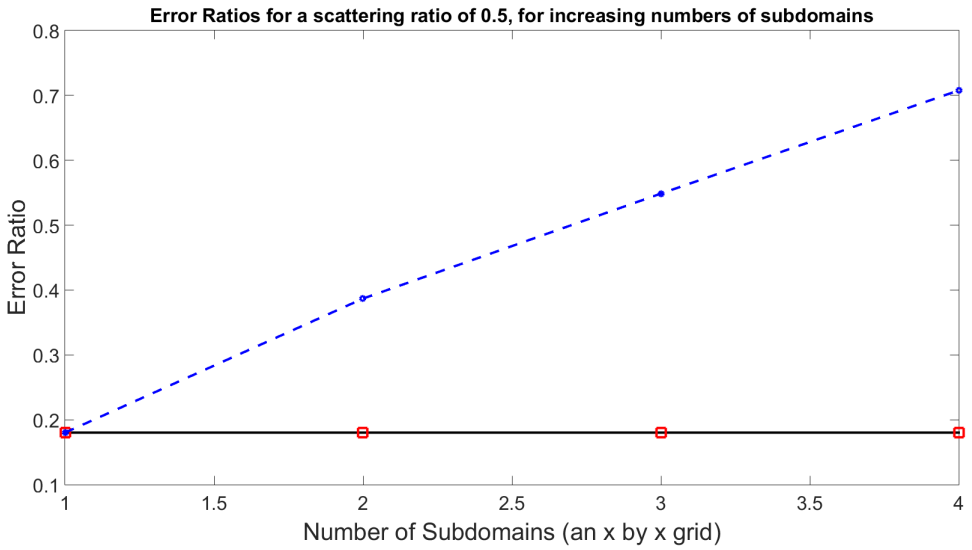

Figure 4-8: Plot of the observed error ratio for Jacobi DDSI (dashed, blue line), Gauss-Seidel DDSI (red boxes, no line) and full SI (black solid line) as the number of subdomains is increased.

that data crossing subdomain boundaries always contains information from the current iteration. In contrast Jacobi DDSI chooses to look at data from the previous iteration, preventing it from converging at the same rate but allowing it to iterate over the subdomains in any order it likes. This means however that Jacobi DDSI requires two full iterations to be stored at all times, a cost that is not incurred by Gauss-Seidel DDSI. Despite the consistently slower rate of convergence of Jacobi DDSI, its higher propensity for parallelisation means that by spreading its computational load over many processors it could ultimately be the fastest method in a practical sense.

As well as varying the scattering ratio, we also varied the number of subdomains that V was decomposed into. Figure 4-8 plots, for a fixed scattering ratio of 0.5, the observed error ratios of each method as the number of subdomains is increased. Clearly full source iteration was not affected by this, and the plot shows that neither was the convergence rate of Gauss-Seidel DDSI, again supporting the conclusion of Theorem 4.4. On the other hand the convergence of Jacobi DDSI was affected, and we see that for higher numbers of subdomains it displays a slower rate of convergence. This makes sense intuitively since Jacobi DDSI's main weakness is the transfer of data between subdomains, and for a higher number of subdomains such data transfer is required more and more. As a final remark we mention that despite exploiting this weakness to its fullest, we were unable to cause Jacobi DDSI to diverge in a situation where full SI (or Gauss-Seidel DDSI) converged. Again, this suggests that the criteria of

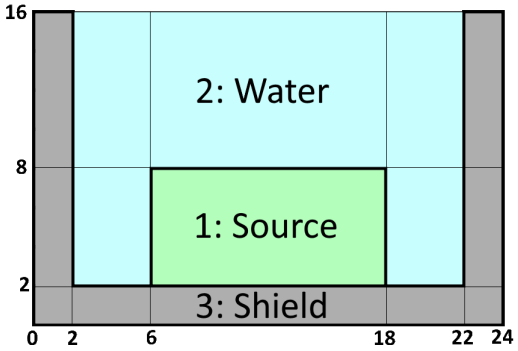

Figure 4-9: Dimensions (in metres) for spent fuel pool physical example.

Theorem 4.8 (that guarantees convergence of Jacobi DDSI) are overly restrictive, and that improvements to this work are very likely to be possible.

4.5.3 Physically Motivated Example: Spent Fuel Pool

To conclude this numerical results section we will consider an example motivated by real-world spent fuel pools. Over time the uranium fuel that powers nuclear reactors gets used up and must be replaced. During this replacement, the fuel assembly that contains the spent fuel is removed and stored for a period of time in a spent fuel pool. These pools are lined with a material that has a high absorption cross section, typically steel, and filled with water. The high scattering cross section of water means that neutrons given off by the spent uranium are very unlikely to reach the surface of the pool since they undergo so many scattering interactions, and consequently lose all their energy before being absorbed.

We will model a 2D cross-sectional representation of a real world pool, with dimensions of 24m by 16m. This pool will have a 2m thick shield along the base and sides, and will contain a 12m by 6m neutron source. A diagram visualising the spatial domain is given in Figure 4-9 and includes dimensions (in metres). We will assume for simplicity that the shield is made entirely of stainless steel, and the cross sectional data we will use is given in Table 4.4. This data was provided by AMEC Foster Wheeler, and is for neutrons at a specific thermal (low) energy. We will assume the cross sections in the source region are the same as those in the shield, as is done in [47], and we will set the source to one so that it is of the same order as the cross sections.

From our theory in Chapter 2 we expect the high scattering ratio of water to cause source iteration to converge very slowly when modelling such a system. Our work in Chapter 3 tells us that using DSA is one method of mitigating against this slow convergence, however it is also a more computationally expensive method per

Material	Cross section (cm/s) Absorption	Scatter
Water	0.02222	3.75922
Stainless Steel	0.24216	0.88442

Table 4.4: Material data for the spent fuel pool physical example. Data provided by AMEC Foster Wheeler for a specific thermal (low) energy.

iteration. If the poor performance of SI is caused by only certain parts of the domain, it would be economical to apply DSA *only* in those parts of the domain and continue to use SI elsewhere.

To model this problem we will decompose the spatial domain into 15 subdomains, indicated by the grid lines in Figure 4-9. We will solve using a hybrid version of our Gauss-Seidel domain decomposition method in which we are able to 'switch on' DSA in any subdomain we like whilst still applying source iteration in the other subdomains (we will refer to this as the Gauss-Seidel SI-DSA algorithm). We will test five different arrangements of these DSA subdomains:

(i) DSA applied in no subdomains, which is equivalent to Gauss-Seidel DDSI (Algorithm 8);

(ii) DSA applied only in subdomains that contain water (region 2);

(iii) DSA applied only in the subdomain containing the source (region 1);

(iv) DSA applied in both the water and source subdomains (regions 1 and 2);

(v) DSA applied in every subdomain.

These numeral identifiers will be referred to in Table 4.5 and Figure 4-10, as well as in the text for the remainder of this section.

Before moving on we will discuss briefly the motivation behind the choices (i)-(v). Firstly, methods (i) and (v) are the two extreme cases and have been included not only so that we can examine their characteristics, but also to act as benchmarks against which the other methods can be compared. Method (ii) applies DSA in only the subdomains where water is present, and will allow us to see that the high scattering ratio of water is not the sole cause of source iterations poor performance in this example. Method (iv) builds upon method (ii) by further applying DSA in the subdomain containing the source. We will see that it is the method that captures most of the convergence behaviour of method (v) where DSA is applied everywhere, but manages to do so at a lower computational cost. Lastly method (iii) sits between methods (ii)

and (iv), and applies DSA only in the region containing the source. This will help us account for the different behaviours we will observe between methods (ii) and (iv).

In this experiment we will run each method for 50 iterations and will measure the error at each iteration with respect to a reference solution as well as the time taken to complete the iterations. This reference solution was obtained by running method (ii) for over 3000 iterations until the residual norm was varying by no more than 10^{-10} between successive iterations. We will solve over a regular triangular mesh of the form in Figure 5-1, with spatial resolutions $M_x = 12$ and $M_y = 8$, and angular resolution $N = 28$. This mesh size causes the element edges to line up exactly with the internal material interfaces of the domain.

We expect that applying DSA in a subdomain will only ever improve upon or maintain the rate of convergence of Gauss-Seidel SI-DSA. However, by measuring the time taken for each version of Gauss-Seidel SI-DSA we should see that applying DSA in more subdomains means the algorithm takes more time per iteration. Our aim is to find that by applying DSA only in those subdomains where it is necessary, we can suffer a minimal loss of convergence rate and yet take a significantly shorter amount of time to converge than if we had applied DSA in every subdomain. This would allow us to draw a balance between the simplicity of source iteration and the power of diffusion synthetic acceleration.

Table 4.5 lists the observed error of each implemented version of Gauss-Seidel SI-DSA, (i)-(v), at all even-numbered iterations. This data is also given graphically in Figure 4-10. Immediately it is clear that all methods perform better than case (i) in which only source iteration is used. This supports the expectation that applying DSA in a subdomain only ever improves the method, however we also see from the remaining four methods that the story is not as simple as more DSA implies faster convergence.

We start by comparing methods (i), no DSA, and (ii), DSA only in region 2; these are the solid (black) line and dotted (blue) line respectively in Figure 4-10. We observe that for the first 20 iterations they perform almost identically, but that the convergence rate of method (i) slows down before that of method (ii). Ultimately both methods suffer this slow-down, however they do so at different values of the error. In fact by about iteration 35, method (ii) has roughly the same error and rate of convergence as method (v) in which DSA was applied in all subdomains. A possible cause of the behaviour exhibited by method (i) is that initially it is resolving errors that exist in non-diffusive regions of the domain, over which it performs reasonably well. However after iteration 20-25 the errors resulting from the diffusive regions dominate and so the pure source iteration method is unable to resolve them quickly, leading to the observed slow down of the convergence ratio. Conversely method (ii) is equipped to handle this

Iteration	Error at each iteration for GS SI-DSA				
	(i)	(ii)	(iii)	(iv)	(v)
2	589.654	591.955	18.363	211.234	212.062
4	362.445	364.048	6.004	7.380	6.641
6	222.817	223.870	4.164	3.299	2.665
8	137.005	137.678	3.253	2.239	1.914
10	84.268	84.679	2.833	1.713	1.566
12	51.863	52.091	2.638	1.465	1.404
14	31.963	32.054	2.534	1.347	1.323
16	19.762	19.741	2.465	1.284	1.275
18	12.313	12.181	2.409	1.242	1.239
20	7.813	7.553	2.359	1.210	1.208
22	5.157	4.739	2.311	1.180	1.180
24	3.660	3.056	2.265	1.153	1.153
26	2.874	2.084	2.221	1.127	1.127
28	2.486	1.556	2.177	1.101	1.101
30	2.295	1.288	2.134	1.077	1.077
32	2.193	1.158	2.092	1.053	1.053
34	2.126	1.090	2.051	1.029	1.029
36	2.075	1.049	2.011	1.006	1.006
38	2.031	1.020	1.972	0.984	0.984
40	1.990	0.995	1.934	0.962	0.962
42	1.951	0.972	1.897	0.941	0.941
44	1.913	0.950	1.860	0.920	0.920
46	1.876	0.929	1.824	0.900	0.900
48	1.840	0.908	1.790	0.880	0.880
50	1.804	0.888	1.755	0.861	0.861

Table 4.5: Table giving the observer error at each iteration when using five different versions of the Gauss-Seidel SI-DSA algorithm (methods (i)-(v)) to solve the spent fuel pool example.

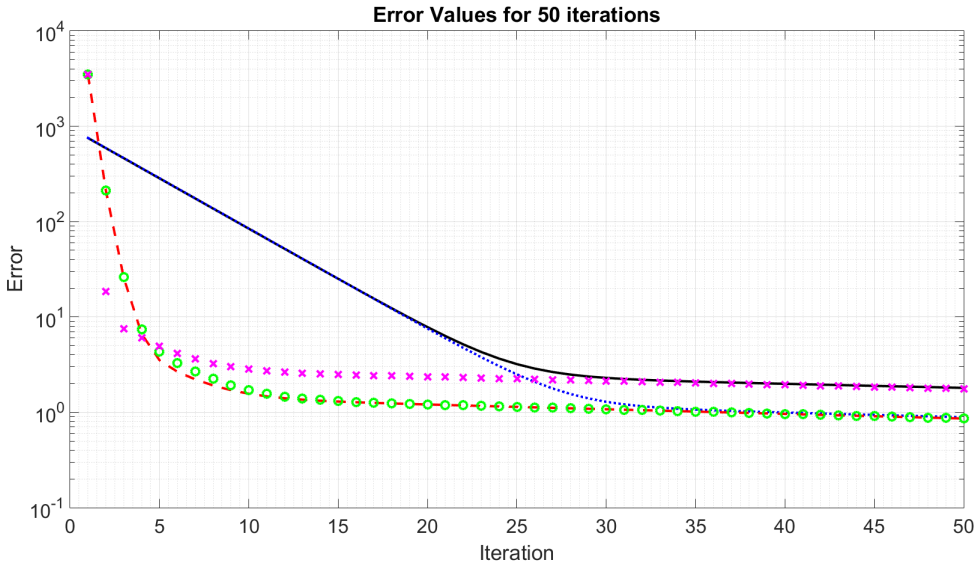

Figure 4-10: Plot of the observed error when running five different versions of the Gauss-Seidel SI-DSA algorithm to solve the spent fuel pool example. Using the numeral identifiers defined in this section, the data presented are (i) black solid line, (ii) blue dotted line, (iii) magenta crosses (no line), (iv) green circles (no line), and (v) red dashed line.

diffusive behaviour and so is able to maintain the higher rate of convergence until it 'catches up' with method (v).

Next we notice that methods (iii), (iv) and (v) exhibit a much faster initial rate of convergence, and a common link between these methods is that they all apply DSA in the subdomain containing the source. In fact method (iii) manages the faster rate of convergence by only applying DSA in this subdomain. To explain this we need to know that the subdomain containing the source also has the highest levels of the neutron flux, which makes sense physically. These high flux values lead to a high value of the error which the faster DSA algorithm is able to resolve more quickly than basic source iteration. Consequently, despite it not being the most diffusive, the source region accounted for much of the error in the domain and so applying a faster method there had the greatest impact on overall convergence rate.

By iteration 10 we start to see method (iii) suffer the same slowing of its rate of convergence as method (i) does by iteration 30. This fits in with our above explanation for method (i) since method (iii) is similarly ill-equipped to handle the diffusive behaviour presented by the water in region 2. By iteration 35 we see that methods (i) and (iii) have roughly the same error and rate of convergence.

Lastly we can compare methods (iv) and (v) and notice that they behave almost

Time in seconds	Version of GS SI-DSA (i)	(ii)	(iii)	(iv)	(v)
per iteration (average)	5.3	10.4	6.5	13.8	16.0
to converge to error = 10^1	100.2	197.8	19.4	55.0	64.0

Table 4.6: Table giving the time (in seconds) taken by the five different versions of the Gauss-Seidel SI-DSA algorithm, (i) - (v), to complete one iteration, and also to converge to an error of 10^1.

identically. This implies that method (iv) has not suffered a reduced rate of convergence by applying DSA only in the water and source regions.

We turn now to Table 4.6 in which we list the average time each method took to complete one iteration, along with the time each method took to resolve the solution to an error less than 10^1. Focussing first on just the 'per iteration' measure we see that as expected method (i) took the shortest time to complete each iteration, however method (iii) was similarly very quick taking only a second longer. The remaining three methods took 2-3 times longer, with method (v) taking the longest.

Next looking at the last row of Table 4.6 we see that despite being quickest per iteration, method (i) was the second slowest to achieve an error below 10^1. The slowest method was (ii), where the inability to quickly resolve the error occuring in the source region meant many more iterations were needed. The three methods which applied DSA in the source region (methods (iii), (iv) and (v)) were the fastest to achieve an error below 10^1. Of these three, method (iii) was the fastest taking only 19 seconds with method (iv) the next fastest taking 36 seconds longer. These three methods were the ones which applied DSA in the source region. This suggests that a good strategy to choose where in the domain to apply DSA might be to focus first on regions which will contribute the most to the error. Once the error in these areas is judged to have reduced far enough, DSA could then be extended to diffusive areas. We did not implement such a method, however it would be an interesting next step.

Conclusions

The observed gain in efficiency was achieved simply by applying the appropriate method to each area of the domain. This is a very general idea, and the domain decomposed algorithms presented in this chapter could also be extended to apply other iterative methods in select subdomains, not just DSA. For example, in [75] Warsa et. al. consider how the effectiveness of multidimensional DSA is degraded by the presence of discontinuous material properties. They note that applying DSA as a preconditioner to a Krylov method overcomes this degradation. It is possible that the additional

cost of this preconditioned Krylov method could be limited, without losing out on its benefits, by applying it only within subdomains that contain discontinuous material properties.

It was also the case that applying a faster method to areas with higher flux greatly improved the rate of convergence. In many cases it is straightforward to predict where 'hotspots' like this might (or will) occur. In these cases, pre-emptively applying a faster method, such as DSA, to subdomains containing those areas could be a simple and economical way of reducing the computational time required for the calculation.

Chapter 5

Finite Element Method for the Steady Neutron Transport Equation

Contents

5.1 Introduction 160

5.2 Discontinuous Galerkin Finite Element Method 161

5.2.1 The Steady Neutron Transport Equation with Constant Angle 161

5.2.2 The Steady Neutron Transport Equation 164

5.2.3 Error Calculation 167

5.1 Introduction

In this chapter we will describe how we discretised the steady 2D transport equation, which comprises of two spatial dimensions and one angular dimension. This will be done using discontinuous Galerkin finite elements in space and discrete ordinates in angle.

A discontinuous finite element scheme is required since using a continuous scheme leads to unphysical oscillations arising in the solution, see for example T. Bennison [13, Section 1.4.2], Buchan et. al. [18], and for a 1D demonstration of these oscillations see [36, Section 1.1]. These oscillations occur for problems containing interior or boundary layers when continuous methods are used with mesh widths too coarse to resolve the layer. One way of overcoming this issue is the so-called *Streamlined Upwind Petrov-Galerkin* method, in which artificial diffusion is encorporated into the

equation to damp out the oscillations [13]. However the approach taken in this thesis is to use a discontinuous Galerkin finite element method to discretise in space, since this method does not require any additional stabilisation, and oscillations are localised only around the boundary layer itself. The main disadvantage to a discontinuous method is the increased number of degrees of freedom which can lead to very large systems of equations.

Discontinuous Galerkin finite element methods (DG FEM) were first developed by Reed and Hill [61] in 1973 for use in solving the steady-state neutron transport equation. The convergence of the method was subsequently analysed in 1974 by LaSaint and Raviart [56], before being improved upon by Johnson and Pitkäranta in 1986 [43]. They showed that for hyperbolic problems with nth degree polynomial basis elements, the DG FEM method has an error of order $\mathcal{O}(h^{n+1/2})$, where h is the mesh width. In practice the better rate of $\mathcal{O}(h^{n+1})$ is often observed [63], though examples can be constructed showing that $\mathcal{O}(h^{n+1/2})$ is sharp (see [58]).

We start in Section 5.2.1 by discretising the transport equation in space, treating angle as a constant. Once this is accomplished, in Section 5.2.2 we allow angle to vary once more and obtain a fully discrete version of the transport equation. Finally in Section 5.2.3 we explain how errors in the approximate scalar flux are calculated in our numerical results sections at the end of Chapters 2, 3 and 4.

To complete this discretisation we referred to the papers mentioned above, as well as Johnson and Pitkäranta, 1983, [42] who give a very clear application of the method to a model problem. We will focus on using linear basis elements ($n = 1$) over a triangular mesh on a rectangular domain.

5.2 Discontinuous Galerkin Finite Element Method

5.2.1 The Steady Neutron Transport Equation with Constant Angle

In this section our aim is to discretise the steady 2D transport equation, as given in (2.8) with zero incoming boundary conditions ((2.9) where $f = 0$). However to begin with we will take angle, Ω, to be constant, and consider how to discretise the equation

$$\Omega \cdot \nabla\psi(\mathbf{r}) + \sigma_T(\mathbf{r})\psi(\mathbf{r}) = g(\mathbf{r}), \tag{5.1}$$

with $\mathbf{r} \in V \subset \mathbb{R}^2$, $g \in L^2(V)$, and $\Omega \in \mathbb{S}^1$ constant. The neutron flux, ψ, is subject to the incoming boundary condition

$$\psi(\mathbf{r}) = 0 \quad \text{if } \Omega \cdot \hat{n}(\mathbf{r}) < 0, \forall \mathbf{r} \in \partial V, \tag{5.2}$$

where $\hat{n}(\mathbf{r})$ is the outward unit normal to ∂V at $\mathbf{r}$.

To solve this we use discontinuous Galerkin finite elements, following the method described in [42] and references therein. We start by dividing the spatial domain into a mesh of triangular *elements*, as shown in Figure 5-1.

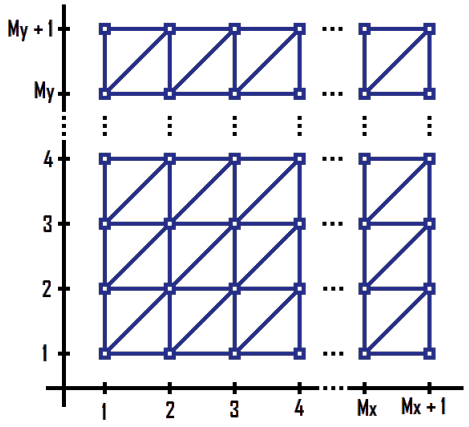

Figure 5-1: Diagram of the spatial mesh that we will apply.

The set of all elements is denoted by $\mathscr{C}_h$, where h specifies the mesh width: the maximum side length of any element. For each element, $\tau \in \mathscr{C}_h$, we define their *inflow* and *outflow* boundaries respectively as

$$\begin{aligned} \partial\tau_- &\equiv \{\mathbf{r} \in \partial\tau \ : \ \hat{n}_\tau(\mathbf{r}) \cdot \Omega < 0\}, \\ \partial\tau_+ &\equiv \{\mathbf{r} \in \partial\tau \ : \ \hat{n}_\tau(\mathbf{r}) \cdot \Omega > 0\}, \end{aligned} \tag{5.3}$$

where $\hat{n}_\tau(\mathbf{r})$ denotes the outward unit normal to τ at $\mathbf{r}$.

We will use the following finite element space

$$\mathcal{V}_h \equiv \{v_h \in L_2(V) \ : \ v_h|_\tau \text{ is linear } \forall \tau \in \mathscr{C}_h\}, \tag{5.4}$$

containing possibly discontinuous piecewise linear functions on V. Before we specify the discrete problem to be solved, we need one final bit of notation. For $\mathbf{r} \in \partial\tau$ we define

$$\psi_h^{\pm}(\mathbf{r}) = \lim_{\epsilon \to 0\pm} \psi_h(\mathbf{r} + \epsilon\Omega), \tag{5.5}$$

and

$$\psi_h^{-}(\mathbf{r}) = 0 \text{ if } \mathbf{r} \in \partial V \text{ with } \hat{n}(\mathbf{r}) \cdot \Omega < 0. \tag{5.6}$$

With this notation we specify our problem by saying that we seek a $\psi_h \in \mathcal{V}_h$ such that

$$\int_\tau (\Omega \cdot \nabla \psi_h + \sigma_T \psi_h) v_h \, \mathrm{d}\mathbf{r} - \int_{\partial \tau_-} (\Omega \cdot \hat{n}_\tau(\mathbf{r})) \left(\psi_h^+ - \psi_h^-\right) v_h \, \mathrm{d}\mathbf{r} = \int_\tau g v_h \, \mathrm{d}\mathbf{r}, \tag{5.7}$$

for all $v_h \in \mathcal{V}_h$ and for all $\tau \in \mathscr{C}_h$.

Next, to form the stiffness matrix and load vector we need to define basis functions on each element. First let us number the corners of each element τ from 1 to 3, starting in the lower left-most corner and moving anticlockwise. We can therefore use the double index τi to refer to the node belonging to element τ and lying in its ith corner, and say that $\mathbf{r}_{\tau i} \in \overline{V}$ is the coordinate of that node. In a similar way, we say that the basis function $\varphi_i^\tau \in \mathcal{V}_h$ is the ith basis function of τ, and is such that

$$\begin{aligned} \varphi_i^\tau(\mathbf{r}_{\tau i}) &= 1, \\ \varphi_i^\tau(\mathbf{r}_{\tau j}) &= 0 \quad j \neq i, \\ \varphi_i^\tau(\mathbf{r}) &= 0 \quad \forall \mathbf{r} \in \hat{\tau} \in \mathscr{C}_h, \text{ where } \hat{\tau} \neq \tau. \end{aligned} \tag{5.8}$$

Substituting these basis functions into the integrals in (5.7) we can define the following matrices

$$T^{\text{int}}_{\tau i, \tau j} \equiv \int_\tau (\Omega \cdot \nabla \varphi_j^\tau) \varphi_i^\tau \, \mathrm{d}\mathbf{r} + \int_\tau \sigma_T \varphi_j^\tau \varphi_i^\tau \, \mathrm{d}\mathbf{r}, \tag{5.9}$$

$$T^{\text{face}}_{\tau i, \tau j} \equiv -\int_{\partial \tau_-} (\Omega \cdot \hat{n}_\tau(\mathbf{r})) \varphi_j^\tau \varphi_i^\tau \, \mathrm{d}\mathbf{r}, \tag{5.10}$$

$$T^{\text{face}}_{\tau i, \hat{\tau} j} \equiv \int_{\partial \tau_-} (\Omega \cdot \hat{n}_\tau(\mathbf{r})) \varphi_j^{\hat{\tau}} \varphi_i^\tau \, \mathrm{d}\mathbf{r}, \tag{5.11}$$

where $\hat{\tau} \neq \tau$, as well as the vector

$$L_{\tau i} \equiv \int_\tau g \varphi_i^\tau \, \mathrm{d}\mathbf{r}, \tag{5.12}$$

for all $\tau, \hat{\tau} \in \mathscr{C}_h$. Next we say that

$$T = \sum_{\tau \in \mathscr{C}_h} \sum_{i,j=1}^{3} \left[T^{\text{int}}_{\tau i, \tau j} + T^{\text{face}}_{\tau i, \tau j} \right] + \sum_{\substack{\tau, \hat{\tau} \in \mathscr{C}_h \\ \tau \neq \hat{\tau}}} \sum_{i,j=1}^{3} T^{\text{face}}_{\tau i, \hat{\tau} j} \tag{5.13}$$

and

$$L = \sum_{\tau \in \mathscr{C}_h} \sum_{i=1}^{3} L_{\tau i}. \tag{5.14}$$

If we define $\mathbb{M} \equiv 3\,|\mathscr{C}_h|$, where $|\mathscr{C}_h|$ denotes the number of elements in $\mathscr{C}_h$, then we

have that $T \in \mathbb{R}^{\mathbb{M}\times\mathbb{M}}$ and $L \in \mathbb{R}^{\mathbb{M}\times 1}$. This leaves us with the matrix-vector system to be solved

$$T\Psi = L, \tag{5.15}$$

where $\Psi \in \mathbb{R}^{\mathbb{M}\times 1}$ is a vector of unknowns whose τith entry approximates $\psi(\mathbf{r}_{\tau i})$.

5.2.2 The Steady Neutron Transport Equation

Up until this point we have treated angle as a constant, whereas in the 2D transport equation it is a variable lying in the 1-sphere: $\Omega \in \mathbb{S}^1$. To discretise in angle we will use discrete ordinates, and divide the angular domain into N discrete points

$$\Omega_1, \ldots, \Omega_N \in \mathbb{S}^1,$$

with associated weights

$$\omega_1, \ldots, \omega_N.$$

These are chosen via an appropriate quadrature rule, and a simple example would be using the trapezoidal rule to approximate a 1D integral. By dividing the domain, say $[a, b] \subset \mathbb{R}$, into N equally spaced points via $a = x_1, \ldots, x_N = b$, we can approximate the integral of a function f over $[a, b]$ using the trapezoidal rule via

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{2}f(x_1) + h\sum_{k=2}^{N-1} f(x_k) + \frac{h}{2}f(x_N),$$

where $h = (b-a)/(N-1)$ is the mesh width. Here the quadrature points are x_i and the quadrature weights are either h or $h/2$.

Different types of quadrature are appropriate for different functions and domains, and a good discussion of appropriate choices of quadrature for the neutron transport problem is given by Johnson and Pitkäranta in [42, Section 4]. For notational completeness, we will denote by U the set of all discrete angles, and by W the set of all associated weights, i.e.

$$\begin{aligned} U &= \{\Omega_1, \ldots, \Omega_N\} \\ W &= \{\omega_1, \ldots, \omega_N\}. \end{aligned} \tag{5.16}$$

Remark 5.1:
Johnson and Pitkäranta note [42, Example 4.1] that a natural choice of quadrature over the 1-sphere is

$$U = \left\{ \cos\left(\frac{2\pi j}{N}\right), \sin\left(\frac{2\pi j}{N}\right) \right\}_{j=1}^{N},$$

with weights $\omega_i = 2\pi/N$ for all $i \in \{1, \ldots, N\}$. It is this set of points and weights that we use for quadrature during our numerical experiments throughout this thesis.

Furthermore, Johnson and Pitkäranta prove convergence of discontinuous Galerkin finite elements applied to the 2D neutron transport equation provided the spatial and angular resolutions are properly related. For the quadrature choice stated above, it is sufficient to impose the condition

$$N = \left\lceil 8\sqrt{\max\{M_x, M_y\}} \right\rceil,$$

as we do for the numerical results throughout this thesis. The factor of 8 is included to ensure a reasonable angular resolution even for coarse spatial grids.

We know that for each angle (i.e. $k = 1, \ldots, N$) we have to solve a system of the form

$$\Omega_k \cdot \nabla\psi(\mathbf{r}, \Omega_k) + \sigma_T(\mathbf{r})\psi(\mathbf{r}, \Omega_k) = g(\mathbf{r}), \tag{5.17}$$

and in our discrete form, we will write this as the matrix-vector equation

$$T^k \Psi^k = L. \tag{5.18}$$

For a known right hand side we can solve this system for each angle Ω_k, $k = 1, \ldots, N$, and obtain the associated approximations Ψ^k to the neutron flux $\psi(\mathbf{r}, \Omega_k)$. However the transport equation we are trying to solve is given by (2.8), and so the right hand side function, $g(\mathbf{r})$, is in fact

$$g(\mathbf{r}) \equiv \sigma_S(\mathbf{r})\phi(\mathbf{r}) + Q(\mathbf{r}), \tag{5.19}$$

where the scalar flux is unknown, but is coupled to the neutron flux by averaging over angle. Note that we are taking the source term to be independent of angle. To include this into our system we start by defining the new matrix

$$S^{\text{int}}_{\tau i, \tau j} \equiv \int_\tau \sigma_S \varphi_j^\tau \varphi_i^\tau \, \mathrm{d}\mathbf{r}. \tag{5.20}$$

as well as the vector

$$\mathbf{Q}_{\tau i} \equiv \int_\tau Q \varphi_i^\tau \, \mathrm{d}\mathbf{r}, \tag{5.21}$$

for all $\tau \in \mathscr{C}_h$. In a similar way to above, we then say that

$$S = \sum_{\tau \in \mathscr{C}_h} \sum_{i,j=1}^{3} S^{\text{int}}_{\tau i, \tau j}, \tag{5.22}$$

and

$$\mathbf{Q} = \sum_{\tau \in \mathscr{C}_h} \sum_{i=1}^{3} \mathbf{Q}_{\tau i}, \tag{5.23}$$

with $S \in \mathbb{R}^{\mathbb{M} \times \mathbb{M}}$ and $\mathbf{Q} \in \mathbb{R}^{\mathbb{M} \times 1}$. With these we can say that for any angle, Ω^k, we are solving the matrix-vector equation

$$T^k \Psi^k - S\Phi = \mathbf{Q}, \tag{5.24}$$

where Φ denotes the discrete scalar flux, such that the τith element of Φ approximates $\phi(\mathbf{r}_{\tau i})$ for all $\tau \in \mathscr{C}_h$, $i = 1, 2, 3$.

To incorporate all angles, we can instead write that we aim to solve the block diagonal system

$$\begin{bmatrix} T^1 & & \\ & \ddots & \\ & & T^N \end{bmatrix} \begin{bmatrix} \Psi^1 \\ \vdots \\ \Psi^N \end{bmatrix} - \begin{bmatrix} S\Phi \\ \vdots \\ S\Phi \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \vdots \\ \mathbf{Q} \end{bmatrix}. \tag{5.25}$$

Lastly, we want to relate the discrete neutron flux and scalar flux via their integral relationship, which in the continuous case is

$$\phi(\mathbf{r}) = \frac{1}{2\pi} \int_{\mathbb{S}^1} \psi(\mathbf{r}, \Omega) \, \mathrm{d}\Omega.$$

We can approximate this integral with a weighted quadrature summation using the weights, ω_k, associated with each angle, Ω^k. Hence we obtain

$$\Phi = \frac{1}{2\pi} \sum_{k=1}^{N} \omega_k \Psi^k.$$

To incorporate this into our block diagonal matrix form of the transport equation, we define the diagonal matrices $W^k \in \mathbb{R}^{\mathbb{M} \times \mathbb{M}}$ to be

$$W^k \equiv \frac{1}{2\pi} \begin{bmatrix} \omega_k & & \\ & \ddots & \\ & & \omega_k \end{bmatrix}, \tag{5.26}$$

for $k = 1, \ldots, N$. Then

$$\Phi = \begin{bmatrix} W^1 & \dots & W^N \end{bmatrix} \begin{bmatrix} \Psi^1 \\ \vdots \\ \Psi^N \end{bmatrix},$$

and so we can rewrite our block diagonal matrix equation, (5.25), to utilise this as follows

$$\left[\begin{array}{ccc|c} T^1 & & & -S \\ & \ddots & & \vdots \\ & & T^N & -S \\ \hline -W^1 & \dots & -W^N & I \end{array}\right] \left[\begin{array}{c} \Psi^1 \\ \vdots \\ \Psi^N \\ \hline \Phi \end{array}\right] = \left[\begin{array}{c} \mathbf{Q} \\ \vdots \\ \mathbf{Q} \\ \hline 0 \end{array}\right]. \tag{5.27}$$

This is a discrete version of the 2D transport equation, and has a similar form to the block operator transport equation seen in Chapter 3, equation (3.112). We note that in practice this matrix is rarely constructed, and instead so-called *matrix free* methods are usually implemented as they are more efficient.

5.2.3 Error Calculation

We conclude this chapter with an explanation of how we calculate the errors in our numerical tests at the end of Chapters 2, 3 and 4. When we refer to the *error* in an approximate solution we mean the L^2-norm of the difference between the approximate and true solutions. We calculate this using a quadrature rule as follows.

We denote by τ_k the kth element of the mesh, and define $|\tau_k|$ to be the area of the kth element. Also, let m_1^k, m_2^k and m_3^k be the three midpoints of the edges of τ_k (see Figure 5-2), and use Φ_A and Φ_T to denote the approximate and true solutions

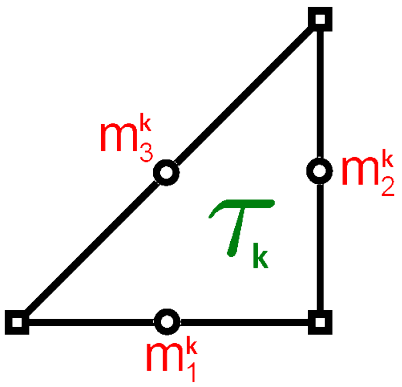

Figure 5-2: Midpoints, m_1^k, m_2^k and m_3^k for a standard mesh element, τ_k.

respectively. Then we calculate the error, $\|\Phi_T - \Phi_A\|_2$, as follows

$$\|\Phi_T - \Phi_A\|_2^2 = \sum_{k=1}^{|\mathscr{C}_h|} \frac{|\tau_k|}{3} \sum_{j=1}^{3} \left(\Phi_T(m_j^k) - \Phi_A(m_j^k)\right)^2.$$

In practice the values of Φ_A and Φ_T at the midpoints are found by assuming they are piecewise-linear on each element, τ, and then extrapolating.

Chapter 6

Summary and Directions for Future Research

In this thesis we have studied the monoenergetic, steady-state neutron transport equation without fission and with isotropic cross sections and isotropic source. Our work was split into three parts:

- Chapter 2: the convergence of source iteration applied to the transport equation,
- Chapter 3: the derivation and application of a diffusion approximation to the scalar flux,
- Chapter 4: the development and convergence of two different domain decomposition source iteration algorithms.

In this concluding chapter we will summarise and review the outcomes of each of these three points of focus.

Firstly in Chapter 2 we derived the solution operator, $\mathcal{K}_{\sigma_T}$, assuming piecewise smooth cross sections in space. We proved that this was compact and positive-definite, before proving a bound on the norm of a scaling of this operator in Theorem 2.14. This result was instrumental in allowing us to prove a new result, Theorem 2.21, confirming the convergence of source iteration when applied to the transport equation with piecewise smooth material data. Theorem 2.21 suggests that the rate of convergence of source iteration is bounded in terms of the maximum norm of the scattering ratio, $\|\sigma_S/\sigma_T\|_\infty$.

Next we proved new bounds on the norm of $\mathcal{K}_{\sigma_T}$ under the assumption of constant cross sections, namely Theorem 2.23 in 2D and 3D, and Theorem 2.25 in 1D. These bounds are tighter than the previous bound in Scheben [67], and they enabled us to

provide a new bound on the rate of convergence of source iteration with constant cross sections in Corollary 2.26. The chapter concluded by numerically verifying our new theory in a 2D setting.

In Chapter 3 we focussed on a specific diffusion equation that can be used to obtain an approximation to the scalar flux. This approximation is well known in the literature, and in 1975 Habetler and Matkowsky [34] used a matched asymptotic expansion method to derive the diffusion approximation and suitable boundary conditions in 1D. Whilst this paper is classical and comprehensive, it carries out a complicated analysis in a relatively short text. Due to this we begin in Section 3.2 by carrying out a thorough matched asymptotic derivation of the diffusion approximation and appropriate boundary conditions, which we think constitutes a useful addition to the existing literature.

In Section 3.3 we outlined general synthetic acceleration algorithms, and stated the well known diffusion synthetic acceleration (DSA) algorithm. We also touched upon how DSA has been formulated as a preconditioner to source iteration, and how this opened the door to applying the same preconditioner to other iterative methods.

Next in Section 3.4 we proved two main results which allowed us to use a novel block operator argument to show the link between the transport equation and the diffusion approximation. Firstly we proved in Theorem 3.5 that the diffusion approximation is a scaling of a Schur complement operator that arises from manipulating a block operator form of the transport equation. Then in Theorem 3.9 we showed that in a certain limit, a scaling of the solution operator $\mathcal{K}_{\sigma_T}$ tends to the identity operator. Together these two results allowed us to conclude the section by deriving (in block operator form) the DSA algorithm from the source iteration algorithm. Theorems 3.5 and 3.9 relied both upon the asymptotic work from Section 3.2 and on several restrictive assumptions. Nonetheless future efforts to make this theory more independent could be very useful in developing our understanding of the diffusion equation and of DSA. We concluded Chapter 3 by exploring the DSA algorithm numerically and comparing its convergence to that of source iteration.

In Chapter 4 we turned our attention to domain decomposition methods. We began in Section 4.2 by specifying two different domain decomposed source iteration (DDSI) algorithms: Jacobi DDSI in Algorithm 6 and Gauss-Seidel DDSI in Algorithm 8. Firstly, in Jacobi DDSI we ensured that when solving over any subdomain, only information from previous iterations on neighbouring subdomains was needed. This method maintains a high potential for parallelisation and allows for the domain to be decomposed into subdomains of any shape. However it also requires data from two iterations to be stored at all times, and results in a slower rate of convergence than

full source iteration. We examined this rate of convergence in Section 4.4 where we derived a convergence criterion for Jacobi DDSI applied over a domain decomposed into two convex subdomains (Theorem 4.8). This criterion was very restrictive, but in Section 4.5 we saw that in practice Jacobi DDSI converged for all tested arrays of many subdomains and for the same range of material data for which Gauss-Seidel DDSI converged.

Conversely, Gauss-Seidel DDSI was focussed towards carrying information through the subdomains during each iteration. It achieved this by solving the subdomains in what we referred to as an *appropriate* order for each angle. Taking care of the flow of information in this way means that Gauss-Seidel DDSI is less suited to parallelisation than Jacobi DDSI. However it only requires the current iteration to be stored at any time, and it maintains all of the convergence behaviour of full source iteration. In fact in Section 4.3, Theorem 4.4, we proved that Gauss-Seidel DDSI applied over a convex domain decomposed into a finite number of convex subdomains was equivalent to the full source iteration algorithm applied over the same domain. This result is important because it tells us that by applying Gauss-Seidel DDSI we are essentially still using full source iteration, however now we have the freedom to treat each subdomain differently. In Section 4.5 we explored this option through the physically motivated example of a spent fuel pool. We saw that using the Gauss-Seidel DDSI algorithm but applying DSA in certain subdomains we were able to greatly reduce the time taken to converge to the same error as either full DSA or full source iteration.

To summarize, in this thesis we have improved the existing convergence theory for the fundamental source iteration algorithm, have extended the range of problems over which it can be proved to converge and tightened the bounds on its rate of convergence. We have given a thorough derivation of the diffusion approximation and associated boundary conditions, and explored a potential new approach to seeing the link between the transport equation and the diffusion approximation. We have also provided two domain decomposition source iteration algorithms with fundamentally different approaches and associated advantages. For these, we have also carried out convergence analysis which can provide a foundation from which the analysis of other domain decomposition algorithms may develop.

We end by providing some potential future research directions that have become clear whilst carrying out this work:

- The convergence analysis in Chapter 2 could be extended to include the effects of fission. Also, more degrees of freedom could be incorporated by allowing for more than one energy group.

- The block operator approach for deriving DSA given in Section 3.4 could be developed by relaxing the smoothness assumptions of Theorem 3.5, allowing it to apply to more physically realistic problems. It could also be made self-contained with less reliance upon the asymptotic work of Habetler and Matkowsky [34] (see Section 3.2). This would allow the approach to be applied outside of the assumptions for diffusivity.

- The convergence theory for Jacobi DDSI could be developed to allow for higher numbers of possibly concave subdomains. Also, the approximations required by this theory could be reduced (or improved upon) so that the resulting criteria for convergence more closely reflect the observed robustness of the Jacobi DDSI method in practice.

- A domain decomposition algorithm that results from combining the approaches of the two domain decomposition algorithms presented here could be investigated. Specifically one could allow the subdomains to be solved in any order, but always use the most up to date information to impose the internal boundary conditions. Such an approach has been implemented in a simple manner before (see [77]) but never thoroughly analysed. It has the potential to maintain the parallelisation options of Jacobi DDSI whilst minimising its loss of convergence rate and reducing the required storage of iterates.

- The preconditioned Krylov DSA method of Warsa et. al. [75] could be combined with the domain decomposition approach outlined in Chapter 4. This might allow subdomains to contain highly discontinuous cross sections without suffering the degraded effectiveness of DSA known to occur in such situations. It could also limit the cost of applying a preconditioned Krylov method over the whole domain.

Appendix A

Appendices

Contents

A.1 Solving a Homogeneous Integro-Differential Equation via Separation of Variables 173

A.2 Fourier Integral Theorem and other results 177

A.1 Solving a Homogeneous Integro-Differential Equation via Separation of Variables

In this section we will consider a homogeneous integro-differential equation given by

$$\frac{\mu}{\hat{\sigma}_T}\frac{\partial}{\partial x}f(\mu,x)+f(\mu,x)=\frac{c}{2}\int_{[-1,1]}f(\mu,x)\,\mathrm{d}\mu, \tag{A.1}$$

where $c>0$ is some scalar. We will see that different behaviour is observed for $c<1$, $c=1$ and $c>1$, and will derive general solutions to this equation for the cases $c=1$ and $c\neq 1$. It is the case when $c=1$ that is of interest in Section 3.2.3. We proceed via separation of variables, and define

$$f\equiv\chi(x)M(\mu). \tag{A.2}$$

Substituting this into (A.1) we can obtain

$$\frac{\frac{\mathrm{d}}{\mathrm{d}x}\chi(x)}{\chi(x)}=\frac{\hat{\sigma}_T c}{2\mu M(\mu)}\int_{[-1,1]}M(\mu)\,\mathrm{d}\mu. \tag{A.3}$$

Setting both sides equal to some separation constant, say $-1/\nu$, we find that

$$\chi(x) = \alpha e^{\frac{-x}{\nu}}, \tag{A.4}$$

so we are searching for solutions of the form

$$f = \alpha M_\nu(\mu) e^{\frac{-x}{\nu}}. \tag{A.5}$$

Here ν is often referred to as an *eigenvalue* with corresponding *eigenfunction*, $M_\nu(\mu)$. Using (A.5) in (A.1) we find that

$$\left[1 - \frac{\mu}{\nu\hat{\sigma}_T}\right] M_\nu(\mu) = \frac{c}{2}\int_{[-1,1]} M_\nu(\mu)\, \mathrm{d}\mu. \tag{A.6}$$

Since $M_\nu(\mu)$ is an eigenfunction, we can normalise it so that

$$\int_{[-1,1]} M_\nu(\mu)\, \mathrm{d}\mu = 1, \tag{A.7}$$

and so (A.6) becomes

$$M_\nu(\mu) = \frac{c}{2}\frac{\nu\hat{\sigma}_T}{\nu\hat{\sigma}_T - \mu} \quad \text{for} \begin{cases} \nu \neq 0, \\ \nu\hat{\sigma}_T \notin [-1,1]. \end{cases} \tag{A.8}$$

We will return to the case when $\nu\hat{\sigma}_T \in [-1,1]$ later. Using (A.7) once more we are left with

$$1 = \frac{c\nu\hat{\sigma}_T}{2} \ln\left[\frac{\nu\hat{\sigma}_T + 1}{\nu\hat{\sigma}_T - 1}\right]. \tag{A.9}$$

Setting $u = \nu\hat{\sigma}_T$, solving (A.9) is equivalent to finding the zeros of

$$\mathcal{V}(u) \equiv 1 - \frac{c}{2} u \ln\left[\frac{u+1}{u-1}\right]. \tag{A.10}$$

Using subscript 0 to denote roots of $\mathcal{V}(u)$, Case and Zweifel [20, Chapter 4, p.63] show that the zeros of $\mathcal{V}(u)$ vary with c in the following manner:

1. $c < 1$: $\mathcal{V}(u)$ has two zeros, $\pm u_0$, on the real axis;
2. $c > 1$: $\mathcal{V}(u)$ has two zeros, $\pm u_0$, on the imaginary axis;
3. $c = 1$: the two roots of $\mathcal{V}(u)$ "coalesce" at ∞.

If $c > 1$ or $c < 1$ for $u \in [-1,1]$, then by (A.8) we know that the eigenfunctions $M_{\pm u_0}(\mu)$ are given by

$$M_{\pm u_0}(\mu) = \frac{c}{2}\frac{u_0}{u_0 \mp \mu}. \tag{A.11}$$

Back in our original notation, noting $\hat{\sigma}_T$ is constant and with $\nu_0 (= u_0/\hat{\sigma}_T)$ denoting roots of (A.9), the eigenfunctions are

$$M_{\pm\nu_0}(\mu) = \frac{c}{2}\frac{\hat{\sigma}_T\nu_0}{\hat{\sigma}_T\nu_0 \mp \mu}. \tag{A.12}$$

Combining this with (A.5) we find the two discrete solutions are given by

$$f_{\pm}(x,\mu) = \frac{c}{2}e^{\frac{\mp x}{\nu_0}}\frac{\hat{\sigma}_T\nu_0}{\hat{\sigma}_T\nu_0 \mp \mu}. \tag{A.13}$$

Cases 1 and 2 are more fully explored in [12, Chapter 2] as well as [20, Section 4.3], where the values that the discrete eigenvalues take are specified.

It is the third case with $c = 1$ that we use in Section 3.2.3, and we will now look at this. Case and Zweifel [20] consider $c = 1$ in Appendix F, where they note that due to having a repeated root we instead must search for solutions to (A.1) of the form

$$f(x,\mu) = m_1(\mu) + m_2(\mu)x, \tag{A.14}$$

where we have noted that

$$\lim_{\nu\to\pm\infty} e^{\frac{-x}{\nu}} = e^{\mp 0} = 1 \mp 0. \tag{A.15}$$

The form of (A.14) can be confirmed by Taylor expanding (A.13) to get

$$f_{\pm}(x,\mu) = \frac{c}{2}\frac{\hat{\sigma}_T\nu_0}{\hat{\sigma}_T\nu_0 \mp \mu}\left[1 \mp \frac{x}{\nu_0} + \frac{x^2}{2\nu_0^2} \mp \ldots\right],$$

then retaining only the first two terms. Substituting (A.14) into the integro-differential equation (A.1) yields

$$\frac{\mu}{\hat{\sigma}_T}m_2(\mu) + m_1(\mu) + xm_2(\mu) = \frac{1}{2}\int_{[-1,1]} m_1(\mu)\,\mathrm{d}\mu + \frac{x}{2}\int_{[-1,1]} m_2(\mu)\,\mathrm{d}\mu, \tag{A.16}$$

and looking at the coefficients of powers of x^0 and x^1 we find

$$x^0: \qquad \frac{\mu}{\hat{\sigma}_T}m_2(\mu) + m_1(\mu) = \frac{1}{2}\int_{[-1,1]} m_1(\mu)\,\mathrm{d}\mu, \tag{A.17}$$

$$x^1: \qquad m_2(\mu) = \frac{1}{2}\int_{[-1,1]} m_2(\mu)\,\mathrm{d}\mu \tag{A.18}$$

From (A.18) we find that m_2 is a constant (independent of μ). If we choose $m_2 = 0$ then (A.17) tells us that m_1 is also a constant, and one simple choice is thus $m_1 = 1$. So we have one solution to (A.1), namely

$$f_a(x, \mu) = 1. \tag{A.19}$$

Alternatively, if we take m_2 nonzero, say $m_2 = \hat{\sigma}_T$, then (A.17) tells us

$$m_1(\mu) = C - \mu, \tag{A.20}$$

where

$$C = \frac{1}{2}\int_{[-1,1]} m_1(\mu)\, \mathrm{d}\mu. \tag{A.21}$$

Since we can form a solution to (A.1) for any C, and taking $C \neq 0$ is equivalent to adding a scaling of the solution (A.19), we take $C = 0$. This leads to

$$f_b(x, \mu) = \hat{\sigma}_T x - \mu. \tag{A.22}$$

The discrete solutions (A.19) and (A.22) are valid when $c = 1$ with $\nu\hat{\sigma}_T \in [-1, 1]$.

We return now to consider the case where $\nu\hat{\sigma}_T \in [-1, 1]$. This allows $\nu\hat{\sigma}_T = \mu$, in which case (A.8) is divergent and will not satisfy (A.7). To resolve this we add an extra term to the right hand side of (A.8) so that it becomes

$$M_\nu(\mu) = \frac{c}{2}\frac{\nu\hat{\sigma}_T}{\nu\hat{\sigma}_T - \mu} + \lambda(\nu)\delta(\nu\hat{\sigma}_T - \mu), \tag{A.23}$$

where λ is an arbitrary function of ν. This still satisfies (A.8) as can be easily verified by substituting it in while recalling that $y\delta(y) \equiv 0$ by definition. If we use the Cauchy principal value method, i.e. if we require

$$\int_{-1}^{1} \frac{1}{\nu\hat{\sigma}_T - \mu}\, \mathrm{d}\mu = \lim_{\tau\to 0}\left[\int_{-1}^{\nu\hat{\sigma}_T - \tau} \frac{1}{\nu\hat{\sigma}_T - \mu}\, \mathrm{d}\mu + \int_{\nu\hat{\sigma}_T + \tau}^{1} \frac{1}{\nu\hat{\sigma}_T - \mu}\, \mathrm{d}\mu\right], \tag{A.24}$$

to be met whenever $M_\nu(\mu)$ is integrated and $\nu\hat{\sigma}_T \in [-1, 1]$, then we are left with the following general form of (A.8)

$$M_\nu(\mu) = \frac{c}{2}P_r\frac{\nu\hat{\sigma}_T}{\nu\hat{\sigma}_T - \mu} + \lambda(\nu)\delta(\nu\hat{\sigma}_T - \mu), \tag{A.25}$$

where P_r denotes the Cauchy principal value. We want to choose the function λ such that (A.25) satisfies (A.7). To do this, we apply the normalisation condition (A.7) to

(A.25) to obtain

$$\lambda(\nu) = 1 - \frac{c\nu\hat{\sigma}_T}{2} P_r \int_{-1}^{1} \frac{1}{\nu\hat{\sigma}_T - \mu} \, \mathrm{d}\mu. \tag{A.26}$$

This principal value integration can be carried out (see [20, Section 4.4]) and it is found that

$$\lambda(\nu) = 1 - c\nu\hat{\sigma}_T \tanh^{-1}(\nu\hat{\sigma}_T) \tag{A.27}$$

Finally substituting (A.25) into (A.5) we get

$$f_\nu(x,\mu) = \alpha e^{\frac{-x}{\nu}} \left[\frac{c}{2} P_r \frac{\nu\hat{\sigma}_T}{\nu\hat{\sigma}_T - \mu} + \lambda(\nu)\delta(\nu\hat{\sigma}_T - \mu) \right] \tag{A.28}$$

which is in fact a continuum of solutions for all $-1 \leq \nu\hat{\sigma}_T \leq 1$.

We now have a set of solutions for $c = 1$ formed by (A.19), (A.22) and (A.28). Case and Zweifel [20] prove that this set is complete (Section 4.6) for assumptions given in their Appendix G. From this set, a general solution can be formed as

$$f(x,\mu) = a + b(\hat{\sigma}_T x - \mu) + \int_{-1/\hat{\sigma}_T}^{1/\hat{\sigma}_T} A(\nu) M_\nu(\mu) e^{\frac{-x}{\nu}} \, \mathrm{d}\nu, \tag{A.29}$$

with $M_\nu(\mu)$ defined by (A.25). To determine the constants a, b and $A(\nu)$ we would require (A.29) to satisfy appropriate boundary conditions also.

Similarly, for $c \neq 0$, the complete set solutions is given by (A.13) and (A.28), and an equivalent general solution can be formed.

A.2 Fourier Integral Theorem and other results

In this section we prove two results given in Section 3.4. Firstly we prove Lemma 3.7, which says the following.

Lemma A.1:
If

$$K(z) \equiv \frac{1}{2} \int_a^b \frac{1}{\mu} \exp\left(\frac{-|z|}{\mu} \right) \, d\mu$$

then

$$\int_{\mathbb{R}} K(z) \, dz = b - a.$$

Proof.

Denoting the Fourier transform of K by $\hat{K}$ we have

$$\hat{K}(\xi) = \int_{-\infty}^{\infty} \exp\left(-2\pi i \xi z\right) K(z)\, \mathrm{d}z. \tag{A.30}$$

With the definition of K this is

$$\hat{K}(\xi) = \int_a^b \int_{-\infty}^{\infty} \frac{1}{2\mu} \exp\left(-2\pi i \xi z\right) \exp\left(\frac{-|z|}{\mu}\right) \mathrm{d}z\, \mathrm{d}\mu.$$

Now evaluating this integral we get

$$\begin{aligned}
\hat{K}(\xi) &= \int_a^b \frac{1}{2\mu} \left[\int_{-\infty}^{0} \exp\left(\frac{z}{\mu} - 2\pi i \xi z\right) \mathrm{d}z + \int_0^{\infty} \exp\left(\frac{-z}{\mu} - 2\pi i \xi z\right) \mathrm{d}z \right] \mathrm{d}\mu \\
&= \int_a^b \frac{1}{2\mu} \left[\frac{\exp\left(\frac{z}{\mu} - 2\pi i \xi z\right)}{\frac{1}{\mu} - 2\pi i \xi} \right]_{-\infty}^{0} + \frac{1}{2\mu} \left[\frac{-\exp\left(\frac{-z}{\mu} - 2\pi i \xi z\right)}{\frac{1}{\mu} + 2\pi i \xi} \right]_{0}^{\infty} \mathrm{d}\mu \\
&= \int_a^b \frac{1}{2\mu} \left[\frac{\mu}{1 - 2\pi i \xi \mu} \right] + \frac{1}{2\mu} \left[\frac{\mu}{1 + 2\pi i \xi \mu} \right] \mathrm{d}\mu \\
&= \int_a^b \frac{1}{1 + (2\pi \xi \mu)^2} \, \mathrm{d}\mu \\
&= \frac{1}{2\pi\xi} \left[\tan^{-1}\left(2\pi\xi b\right) - \tan^{-1}\left(2\pi\xi a\right) \right].
\end{aligned}$$

Taking a taylor expansion results in

$$\begin{aligned}
\hat{K}(\xi) &= \frac{1}{2\pi\xi} \left(2\pi\xi b - 2\pi\xi a - \frac{(2\pi\xi b)^3}{3} + \frac{(2\pi\xi a)^3}{3} + \dots \right) \\
&= (b - a) + O(\xi^2).
\end{aligned}$$

and so we have that $\hat{K}(0) = (b - a)$. Using this with (A.30) yields the result. □

Next we will prove Lemma 3.8, which says the following.

Lemma A.2:
Let $f : \mathbb{R} \to \mathbb{R}$, and suppose that f is Lipschitz continuous and also globally bounded so that $\max_\xi |f(\xi)| = C$, where C is constant. Suppose also we have some function $K : \mathbb{R} \to \mathbb{R}^+$ such that

$$\int_{\mathbb{R}} K(z)\, dz = \alpha,$$

where $\alpha \in \mathbb{R}$ is some constant. Then it holds that

$$\lim_{\sigma_T \to \infty} \sigma_T \int_{\mathbb{R}} K\left(\sigma_T (x-y)\right) f(y)\ dy = \alpha f(x), \tag{A.31}$$

for all $x \in \mathbb{R}$.

Proof.
First of all we set $\delta \equiv {\sigma_T}^{-1}$ (so we are considering the limit as $\delta \to 0$) and separate the integral in (A.31) into two parts, leaving

$$\frac{1}{\delta}\int_{\mathbb{R}} K\left(\delta^{-1}(x-y)\right) f(y)\ \mathrm{d}y = \overbrace{\frac{1}{\delta}\int_{\mathbb{R}} K\left(\delta^{-1}(x-y)\right) f(x)\ \mathrm{d}y}^{\equiv I_1} + \underbrace{\frac{1}{\delta}\int_{\mathbb{R}} K\left(\delta^{-1}(x-y)\right)\left[f(y)-f(x)\right]\mathrm{d}y}_{\equiv I_2}.$$

We will tackle these independently. First of all we consider I_1, and apply a change of variables, $z \equiv \delta^{-1}(x-y)$.

$$\begin{aligned} I_1 &= \frac{f(x)}{\delta}\int_{-\infty}^{\infty} K\left(\delta^{-1}(x-y)\right)\ \mathrm{d}y \\ &= \frac{f(x)}{\delta}(-\delta)\int_{\infty}^{-\infty} K\left(z\right)\ \mathrm{d}z \\ &= \alpha f(x), \end{aligned}$$

since $\int_{\mathbb{R}} K(z)\,\mathrm{d}z = \alpha$. It therefore remains to prove that $I_2 \to 0$ as $\delta \to 0$. To do this we further separate the remaining integral as follows

$$I_2 = \overbrace{\frac{1}{\delta}\int_{[x-\delta^{\frac{1}{2}},x+\delta^{\frac{1}{2}}]} K\left(\delta^{-1}(x-y)\right)\left[f(y)-f(x)\right]\mathrm{d}y}^{\equiv I_{21}} + \underbrace{\frac{1}{\delta}\int_{\mathbb{R}\setminus[x-\delta^{\frac{1}{2}},x+\delta^{\frac{1}{2}}]} K\left(\delta^{-1}(x-y)\right)\left[f(y)-f(x)\right]\mathrm{d}y}_{\equiv I_{22}},$$

and show that each of I_{21} and I_{22} tends to zero with δ. Considering I_{21} first, we can use the Lipschitz continuity of f (with constant L) to obtain

$$|I_{21}| = \frac{1}{\delta}\int_{x-\delta^{\frac{1}{2}}}^{x+\delta^{\frac{1}{2}}} \left|K\left(\delta^{-1}(x-y)\right)\right| |f(y)-f(x)| \ \mathrm{d}y$$
$$\leq \frac{L}{\delta}\int_{x-\delta^{\frac{1}{2}}}^{x+\delta^{\frac{1}{2}}} K\left(\delta^{-1}(x-y)\right) |y-x| \ \mathrm{d}y.$$

Noting that $|y-x| \leq \delta^{\frac{1}{2}}$ and applying the same change of variables as before we get

$$|I_{21}| \leq \frac{L\delta^{\frac{1}{2}}}{\delta}(-\delta)\int_{\delta^{-\frac{1}{2}}}^{-\delta^{-\frac{1}{2}}} K(z) \ \mathrm{d}z$$
$$= L\delta^{\frac{1}{2}}\int_{-\delta^{-\frac{1}{2}}}^{\delta^{-\frac{1}{2}}} K(z) \ \mathrm{d}z.$$

Since $\delta^{\frac{1}{2}} \to 0$ as $\delta \to 0$, we have that $I_{21} \to 0$ also.

Now to show that $I_{22} \to 0$ with δ we split our integral one last time as

$$I_{22} = \overbrace{\frac{1}{\delta}\int_{x+\delta^{\frac{1}{2}}}^{\infty} K\left(\delta^{-1}(x-y)\right)[f(y)-f(x)] \ \mathrm{d}y}^{\equiv I_{22}^{+}} + \underbrace{\frac{1}{\delta}\int_{-\infty}^{x-\delta^{\frac{1}{2}}} K\left(\delta^{-1}(x-y)\right)[f(y)-f(x)] \ \mathrm{d}y}_{\equiv I_{22}^{-}}.$$

Using the global maximum of f and applying the same change of variables, we proceed as follows

$$\left|I_{22}^{+}\right| \leq \frac{2C}{\delta}\int_{x+\delta^{\frac{1}{2}}}^{\infty} K\left(\delta^{-1}(x-y)\right) \ \mathrm{d}y$$
$$= \frac{2C}{\delta}(-\delta)\int_{-\delta^{-\frac{1}{2}}}^{-\infty} K\left(z\right) \ \mathrm{d}z$$
$$= 2C\int_{-\infty}^{-\delta^{-\frac{1}{2}}} K(z) \ \mathrm{d}z.$$

Since $-\delta^{-\frac{1}{2}} \to -\infty$ as $\delta \to 0$, this tells us that $I_{22}^{+} \to 0$ with δ too.

Similarly for I_{22}^{-} we find that

$$\left|I_{22}^{-}\right| \leq 2C\int_{\delta^{-\frac{1}{2}}}^{\infty} K(z) \ \mathrm{d}z,$$

which also goes to zero with δ. Thus the result holds. □

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, 1972.

[2] M. L. Adams and E. W. Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in Nuclear Energy*, 40(1):3 – 159, 2002.

[3] M. L. Adams and W. R. Martin. Diffusion synthetic acceleration of discontinuous finite element transport iterations. *Nuclear Science and Engineering*, 111(2):145–167, 1992.

[4] R. E. Alcouffe. A stable diffusion synthetic acceleration method for neutron transport iterations. *Trans. Am. Nucl. Soc.*, 23:203, 1976.

[5] R. E. Alcouffe. Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations. *Nucl. Sci. Eng.;(United States)*, 64(2):344, 1977.

[6] S. F. Ashby, P. N. Brown, M. R. Dorr, and A. C. Hindmarsh. Preconditioned iterative methods for discretized transport equations. *Proc. International Topical Meeting on Advances in Mathematics, Computations, Reactor Physics*, 2:6.1 2–1, 1991.

[7] S. F. Ashby, P. N. Brown, M. R. Dorr, and A. C. Hindmarsh. A linear algebraic analysis of diffusion synthetic acceleration for the Boltzmann transport equation. *SIAM J. Numer. Anal.*, 32:128–178, 1995.

[8] Y. Azmy. On the adequacy of message-passing parallel supercomputers for solving neutron transport problems. *Proc. Supercomputing '90*, page 693, 1990.

[9] Y. Azmy. Impossibility of unconditional stability and robustness of diffusive acceleration schemes. *Proc. Topl.Mtg. 1998 American Nuclear Society, Radiation Protection and Shielding Division, Nashville, Tennesse, April 19-23*, 1:480, 1998.

[10] Y. Azmy, T. Wareing, and J. Morel. Effect of material heterogeneity on the performance of DSA for even-parity S_N methods. *International Conference on Mathematics and Computation, Reactor Physics, and Environmental Analysis in Nuclear Applications*, 1:55–63, 1999.

[11] Yousry Azmy and Enrico Sartori. *Nuclear Computational Science: A Century in Review*. Springer, 2010.

[12] G. I. Bell and S. Glasstone. *Nuclear Reactor Theory*. Van Nostrand Reinhold Company, 1970.

[13] T. A. J. Bennison. *Adaptive Discontinuous Galerkin Methods for the Neutron Transport Equation*. PhD thesis, University of Nottingham, 2014.

[14] M. Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182(2):418–477, 2002.

[15] B. L. Bihari and P. N. Brown. A linear algebraic analysis of diffusion synthetic acceleration for the Boltzmann transport equation II: The simple corner balance method. *SIAM J. Numer. Anal.*, 47:1782–1826, 2009.

[16] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods (Texts in Applied Mathematics)*. Springer, 2008.

[17] P. N. Brown. A linear algebraic development of diffusion synthetic acceleration for three-dimensional transport equations. *SIAM J. Numer. Anal.*, 32:179–214, 1995.

[18] A. G. Buchan, A. S. Candy, S. R. Merton, C. C. Pain, J. I. Hadi, M. D. Eaton, A. J. H. Goddard, R. P. Smedley-Stevenson, and G. J. Pearce. The inner-element subgrid scale finite element method for the Boltzmann transport equation. *Nuclear Science and Engineering*, 164(2):105–121, 2010.

[19] D. G. Cacuci, editor. *Handbook of Nuclear Engineering, (5 Volume set)*. Springer, 2010.

[20] K. M. Case and P. F. Zweifel. *Linear transport theory*. Addison-Wesley series in nuclear engineering. Addison-Wesley Pub. Co., 1967.

[21] Tony F. Chan and Tarek P. Mathew. Domain decomposition algorithms. In *Acta Numerica 1994*, pages 61–143. Cambridge University Press, 1994.

[22] S. Chandrasekhar. *Radiative Transfer*. Dover Publications, 1960.

[23] C. Clapham and J. Nicholson. *The Concise Oxford Dictionary of Mathematics (Oxford Paperback Reference)*. Oxford University Press, 2009.

[24] J. J. Duderstadt and W. R. Martin. *Transport Theory*. John Wiley and Sons, New York, 1979.

[25] J. D. Evans. Private Communication. University of Bath.

[26] T. M. Evans, S. W. Mosher, S. R. Slattery, and S. P. Hamilton. A monte carlo synthetic-acceleration method for solving the thermal radiation diffusion equation. *Journal of Computational Physics*, 258(0):338 – 358, 2014.

[27] V. Faber and T. A. Manteuffel. A look at transport theory from the point of view of linear algebra. *Conference: Transport theory, invariant imbedding and integral equations*, 1988.

[28] E. M. Gelbard. Application of spherical harmonics methods to reactor problems. *Bettis Atomic Power Laboratory*, 1960.

[29] F. Golse, S. Jin, and C. D. Levermore. The convergence of numerical transfer schemes in diffusive regimes I: Discrete-ordinate method. *SIAM Journal on Numerical Analysis*, 36(5):1333–1369, 1999.

[30] J. Gonçalves and P. J. Coelho. Parallelization of the discrete ordinates method. *Numerical Heat Transfer Part B-Fundamentals*, 32(2):151–173, 1997.

[31] I. G. Graham and I. H. Sloan. On the compactness of certain integral operators. *Journal of Mathematical Analysis and Applications*, 68(2):580 – 594, 1979.

[32] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, 1997.

[33] C. Grossmann, Hans-G. Roos, and M. Stynes. *Numerical Treatment of Partial Differential Equations (Universitext)*. Springer, 2007.

[34] G. J. Habetler and B. J. Matkowsky. Uniform asymptotic expansions in transport theory with small mean free paths, and the diffusion approximation. *J. Math. Phys*, 1975.

[35] L. A. Hageman and D. M. Young. *Applied Iterative Methods*. Academic Press, inc., 1981.

[36] E. Hall. *Anisotropic Adaptive Refinement For Discontinuous Galerkin Methods*. PhD thesis, University of Leicester, 2007.

[37] E. J. Hinch. *Perturbation Methods (Cambridge Texts in Applied Mathematics)*. Cambridge University Press, 1991.

[38] H. Hochstadt. *Integral equations*. Wiley, 1973.

[39] I. C. F. Ipsen and C. D. Meyer. The idea behind Krylov methods. *Amer. Math. Monthly*, 105(10):889–899, 1998.

[40] S. Jin and C. D. Levermore. The discrete-ordinate method in diffusive regimes. *Transport Theory and Statistical Physics*, 20(5&6):413–439, 1991.

[41] S. Jin and C. D. Levermore. Fully-discrete numerical transfer in diffusive regimes. *Transport Theory and Statistical Physics*, 22(6):739–791, 1993.

[42] C. Johnson and J. Pitkäranta. Convergence of a fully discrete scheme for two-dimensional neutron transport. *SIAM Journal on Numerical Analysis*, 20(5):951–966, 1983.

[43] C. Johnson and J. Pitkäranta. An analysis of the discontinuous galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 46(173):1–26, 1986.

[44] L. V. Kantorovich and G. P. Akilov. *Functional Analysis*. Pergamon Press, 1982.

[45] H. G. Kaper and R. B. Kellogg. Continuity and differentiability properties of the solution of the linear transport equation. *SIAM Journal on Applied Mathematics*, 32(1):201–214, 1977.

[46] J. P. Keener. *Principles Of Applied Mathematics (Advanced Book Program)*. Westview Press, 2000.

[47] K. Kobayashi, N. Sugimura, and Y. Nagaya. 3-D radiation transport benchmark problems and results for simple geometries with void regions. Technical report, Nuclear Energy Agency, 2000.

[48] H. J. Kopp. Synthetic method solution of the transport equation. *Nucl. Sci. Eng.*, 17:65, 1963.

[49] E. W. Larsen. Solutions of the steady, one-speed neutron transport equation for small mean free paths. *J. Mathematical Phys.*, 15:299–305, 1974.

[50] E. W. Larsen. Diffusion theory as an asymptotic limit of transport theory for nearly critical systems with small mean free paths. *Annals of Nuclear Energy*, 7(4-5):249 – 255, 1980.

[51] E. W. Larsen. Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. Part I: Theory. *Nucl. Sci. Eng.*, 82:47–63, 1982.

[52] E. W. Larsen. Diffusion-synthetic acceleration methods for discrete ordinates problems. *Trans. Thry. and Stat. Phys.*, 13:107–126, 1984.

[53] E. W. Larsen and J. B. Keller. Asymptotic solution of neutron transport problems for small mean free paths. *Journal of Mathematical Physics*, 15(1):75–81, 1974.

[54] E. W. Larsen, J. E. Morel, and W. F. Miller Jr. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes. *Journal of Computational Physics*, 69(2):283 – 324, 1987.

[55] E. W. Larsen, J. E. Morel, and J. M. McGhee. Asymptotic derivation of the multigroup P1 and simplified PN equations with anisotropic scattering. *Nuclear science and engineering*, 123(3):328–342, 1996.

[56] P. Lasaint and P. A. Raviart. On a finite element method for solving the neutron transport equation. In *Mathematical aspects of finite elements in partial differential equations (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1974)*, pages 89–123. Publication No. 33. Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York, 1974.

[57] E. E. Lewis and W. F. Miller. *Computational Methods of Neutron Transport*. John Wiley and Sons, Inc., New York, NY, 1984.

[58] T. E. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal on Numerical Analysis*, 28(1):133–140, 1991.

[59] A. K. Prinja and E. W. Larsen. *Handbook of Nuclear Engineering, (5 Volume set)*, chapter 5. General Principles of Neutron Transport. Volume I of Cacuci [19], 2010.

[60] W. H. Reed. The effectiveness of acceleration techniques for iterative methods in transport theory. *Nucl. Sci. Eng.*, 45:245–254, 1971.

[61] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. *Proceedings of the American Nuclear Society*, 1973.

[62] L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the

stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210:307–357, 1911.

[63] G. Richter. On the order of convergence of the discontinuous Galerkin method for hyperbolic equations. *Mathematics of Computation*, 77(264):1871–1885, 2008.

[64] F. Riesz and B. SZ.-Nagy. *Functional Analysis.* Frederick Ungar Publishing co., 1955.

[65] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*, volume 158. SIAM, 1992.

[66] Y. Saad. *Iterative Methods for Sparse Linear Systems, Second Edition.* Society for Industrial and Applied Mathematics, 2003.

[67] F. Scheben. *Iterative Methods for Criticality Computations in Neutron Transport Theory.* PhD thesis, University of Bath, 2011.

[68] H. A. Schwarz. Über einen grenzübergang durch alternierendes verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, May 1870.

[69] W. M. Stacey. *Nuclear Reactor Physics.* Wiley-VCH, 2007.

[70] J. H. Tait. *Neutron Transport Theory.* Longmans, Green and co. LTD, 1964.

[71] T. J. Urbatsch. *Iterative Acceleration Methods for Monte Carlo and Deterministic Criticality Calculations.* PhD thesis, University of Michigan, 1995.

[72] T. A. Wareing, E. W. Larsen, and M. L. Adams. Diffusion accelerated discontinuous finite element schemes for the S_N equations in slab and X-Y geometries. *Proc. International Topical Meeting on Advances in Mathematics, Computations, Reactor Physics*, 3, 1991.

[73] J. S. Warsa, T. A. Wareing, and J. Morel. Fully consistent diffusion synthetic acceleration of linear discontinuous S_N transport discretizations on unstructured tetrahedral meshes. *Nuclear Science and Engineering*, 141(3):236–251, 2002.

[74] J. S. Warsa, T. A. Wareing, and J. E. Morel. Krylov iterative methods applied to multidimensional S_N calculations in the presence of material discontinuities. *Nuclear Mathematical and Computational Sciences: A century in Review, A Century Anew*, 2003.

[75] J. S. Warsa, T. A. Wareing, and J. E. Morel. Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional S_N calculations in problems with material discontinuities. *NSE*, pages 218–248, 2004.

[76] M. Yavuz and E. W. Larsen. Spatial domain decomposition for neutron transport problems. *Transport Theory and Statistical Physics*, 18(2):205–219, 1989.

[77] M. Yavuz and E. W. Larsen. Iterative methods for solving x-y geometry S_N problems on parallel architecture computers. *Nucl. Sci. Eng.*, 112:32–42, 1992.